Deliverable 4.3 Content and Concept Filter V1

Rüdiger Klein, Manuel Kober, Jingquan Xie / Fraunhofer

Dorothea Tsatsou / CERTH

Sacha Weinberg, Jan Thomsen / CONDAT

17/10/2012

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	*17th October 2012*
Deliverable number	*D4.3*
Deliverable name	*Content and Concept Filter V1*
File	*LinkedTV_D4.3.pdf*
Nature	*Report*
Status & version	*Revision: Version 1.0*
Number of pages	*80*
WP contributing to the deliverable	*4*
Task responsible	*Fraunhofer*
Other contributors	*CERTH, CONDAT*
Author(s)	*Rüdiger Klein, Manuel Kober, Jingquan Xie / Fraunhofer* *Dorothea Tsatsou / CERTH* *Sacha Weinberg, Jan Thomsen / CONDAT*
Reviewer	*Rolf Fricke, Condat*
EC Project Officer	*Thomas Kuepper*
Keywords	*Semantic User Models and Content Filtering*
Abstract (for dissemination)	*(see below)*

[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))

Table of contents

List of Figures

1 Abstract

LinkedTV is dedicated to the widespread and rich domains occurring in multimedia content on the Web. In such rich domains it is essential for the users to get support in finding the kind of content they are interested in and to make use of the rich relations between multimedia items on the Web. User models are used to represent the different kinds of interests people may have in multimedia content.

In this document we describe how a user model (introduced in deliverable D4.2) can be used *to filter* multimedia content in various ways and *to support* the user in this way to manage the large amount of multimedia information available on the Web.

A user model contains two main aspects: a description of the user himself (age, profession, social status, etc.), and a representation of those things in the world he is interested in.

Whereas his personal description results in a set of data the representation of his interests needs a more complex form. User interests typically cover a broad spectrum of topics represented in a user model ontology (LUMO). It represents the mental model of the user, i.e., the main concepts, topics, concrete entities, and semantic relationships between them he maintains about the world.

The entities in this user model ontology are related to items in various LOD ontologies like DBPedia, schema.org, the music ontology, etc. This enables us to use the LOD universe as semantic background for user modelling.

The different degrees of interest a user has in various topics are represented as weights for each element in the user model ontology.

The semantic annotation process in LinkedTV enables fine grained annotations of media *fragments* (see the LinkedTV deliverables D2.2 and D2.3). A video as a whole as well as scenes in it or even single shots can be annotated. The multimedia fragments are annotated with elements from LOD ontologies (URI) like DBPedia, music ontology, etc. They are interlinked to other entities on the Web.

Our content filtering is based on weighted semantic matching. It can be used in different ways: enriching information about an object shown in a video scene or frame with linked information from the Web; ranking annotation elements occurring in a frame according to the user's special interest; or determining semantic similarity between media fragments and providing user recommendations.

Six concrete user models are described in this document in order to illustrate our approach showing how different user interests can be and what it meaans for their media consumption.

A first version of the LinkedTV semantic filter LSF has been implemented. It takes semantic user models and semantically enriched media fragment annotations to compute rankings of media content w.r.t. specific user interests. Additionally, we show how a logical reasoner (f-PocketKRHyper developed by our Partner CERTH) can be used with its logic based user model components to post-process the filtering results by using fuzzy logic reasoning.

2 Introduction

In recent years, there has been a lot of research on semantic content representations in multimedia on the Web (see chapter 3 for a more detailed state of the art description). LinkedTV's mission is to extend these semantic representations to a more fine grained scale of *media fragments* including scenes or even single shots. Each of them will be semantically annotated allowing us to present multimedia content in more detail. In order to manage such rich content descriptions we want to support users in various ways. In order to provide this support we use user models capturing the main characteristics of the user and his concrete interests.

In deliverable D4.2 we described the LinkedTV approach to user models: their basic structure and how they are created and maintained. In this document D4.3 we show how these user models can be used in content filtering. Content filtering will allow the LinkedTV system to support the user to manage the huge amount of information in multimedia repositories.

A user model contains two main aspects: a description of the user himself (age, profession, social status, etc.), and a representation of those things in the world he is interested in.

Describing the user as a person results in a set of data about his age, location, profession, etc. The representation of his interests needs a more complex form. User interests typically cover a broad spectrum of topics which are semantically related to other topics. That's the reason why our user model is based on a user model ontology (LUMO). It represents the mental model of the user, i.e., the main concepts, topics, concrete entities, and semantic relationships between them he maintains about the world.

A user model ontology contains the notions the user applies to describe his view on the world, their (main) relationships, and a set of concrete entities (like places, people, events, etc.) he is interested in. The main relationships we are currently using in LinkedTV are subclass relations between concepts, type relations between concrete entities (instances) and their concepts, and a set of domain specific relations between concrete entities (like 'participates' connecting people and events, or 'knows' relating people to other people).

The degree of interest a user has in a certain concept or instance can be represented as a weight. We choose to take real numbers to represent these weights.

The other side of the medal is the multimedia object – in particular videos in LinkedTV. Whereas general video annotation has been dealt with in various approaches (see chapter 3) LinkedTV extends this research in two directions:

- LinkedTV multi media fragment annotations provide a much more fine grained description of multimedia content. Not just the video as a whole but also its scenes and even single shots and frames are annotated and can be used for content filtering.
- The annotations are semantically enriched by relations to other information available on the LOD Web [Anton08].

A user model allows us to find out how much a user is interested in a specific content described in this way. If in a single frame a building is shown – as main content element in this frame or just as background for some scene – the user may find this building interesting and wants to find out more about it. He may not know what an object it is and wants to learn this. Alternatively, if he knows already what a building it is he is interested in enriched information about it: its history, or its current usage, or its architecture, etc. The semantic user model helps us to discriminate between these different cases and to provide the user with a personalized multimedia experience.

The video and its various media fragments are annotated with a set of URI "tags" describing its content – as a whole or in certain parts of it (scenes, shots, etc. – see D2.2). Relating these URI tags with their meaning to items in the user model is the main task we have to deal with in semantic filtering.

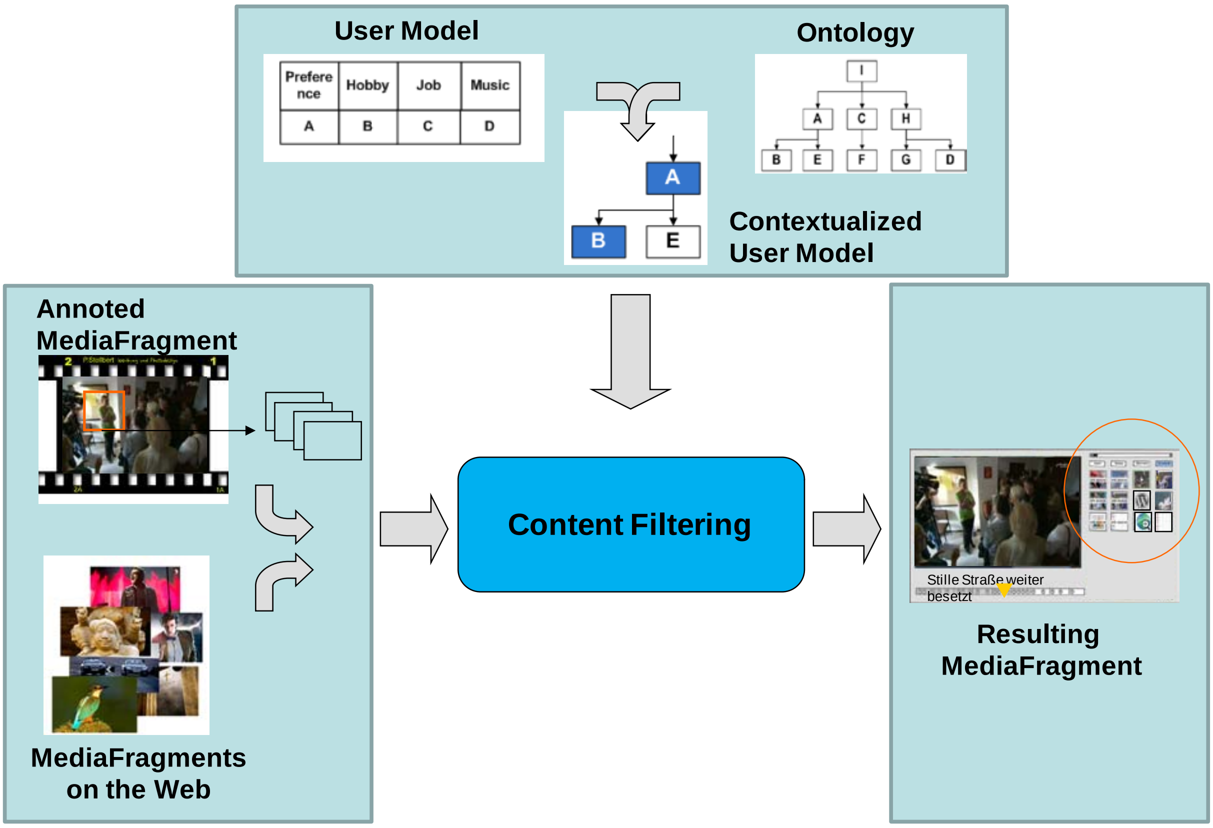


Figure 1: The user model is used to filter annotated media fragments and enrich them with multimedia content on the Web through semantic relations.

This is typically *not* a *one-to-one* relationship where we just have to match tags in the multimedia fragment annotation to elements in the user model. These tags represent the meaning of the video content through their *semantic* relationships to the elements in the user model. These semantic relationships are represented in a certain ontology (like DBPedia, MusicOntology, etc.) and allow us to use more general descriptions of interest in the user model (like architecture or history) and relate them to concrete entities the video or its fragments are annotated with.

Consequently, we have to consider three issues and their relationships for content filtering (see Figure 1):

- the user model (described in D4.2)
- the semantic media fragment annotations (described in D2.2) and
- the semantic mappings between them (described in this document).

The challenge in LinkedTV is that all these activities (and those in the other work packages like use case modeling, user interaction capabilities, and the integration platform) are tightly related to each other. Progress in WP2 on multimedia annotations will help us in WP4 with user modeling and filtering, more sophisticated use case models will help us to identify user needs in content filtering, and the characteristics of the filtering can be specified more precisely if appropriate user interactions are available. In this document we introduce a first approach to semantic content filtering based on the achievements in the other WPs and tasks.

This document is structured as follows: in the next chapter we describe related research and summarize the state of the art in user modeling. In the following chapter 4 we outline what an ontology should look like to be used as basis for a user model in a semantically rich multimedia world. In chapter 5 we describe the methodology used to build the user model ontology UMO. In Chapter 6 we define how to get a user model UM and a contextualized user model CUM based on the LinkedTV user model ontology LUMO. Chapter 7 shows how the CUM is used in content filtering. Chapter 8 deponstrates our approach with examples from our LinkedTV use cases (WP6). In chapter 9 we discuss some efficiency aspects and their relations to modeling. In chapter 10 we introduce our LinkedTV Semantic Filtering tool LSF, followed in chapter 11 by a description of CERTH's logical reasoner f-PocketKRHyper to be used as post-filter processor. We conclude in chapter 12 with a summary and an outlook to the research to follow in Year-2 of the project.

3 Related Work & State of the Art

3.1 Overview

User Models and their application for recommendations and guidance have a long tradition in computer science [Heck05, Jann12, Ricci11, Yoo12]. The more complex the information to be dealt with is the more does a user need support to manage it [Sieg07]. Because users are quite different in their intentions, goals, skills, etc. the support provided by the system should be adapted as much as possible to the concrete user. The history of user models and their application can be seen as a continuous increase in user model expressiveness, precision, and adaptability.

In the following we want to give a brief overview on the state of the art of these three main issues in user models - expressiveness, precision, and adaptability. This document D4.3 is focused on applying user models on content filtering in order to guide the user through the huge information spaces. It is strongly correlated with the “sibling document” D4.2 on user modeling and user model adaptation. This state of the art chapter focuses on *applying* user models on content filtering in relation to the kind and expressiveness of the user models and the way they are built and maintained.

How expressive user models have to be strongly depends on the domain of discourse and the purpose of the user models in them. Some domains show a relatively simple structure: if, for instance, the user model has to support the user in selecting the right command in order to perform a certain task the user mode has to maintain the knowledge about commands to be used for certain tasks and the knowledge the user has acquired about these commands and the related tasks. The domain model may be a simple relation between commands and tasks, and the user model may show how well a user has maintained this relation for a given command and/or a task at hand.

If the purpose of the user model is to guide a user in selecting books in a book store information has to be maintained about how books are related to each other (“similarity”) and about the preferences of this user in various topics shown in recent purchases. It strongly depends on the granularity and precision needed for these recommendations how detailed the domain model and the corresponding user model have to be. If the decision has to be made just on categories like thrillers, love stories, and historic fiction a simple classification of books and of user preferences is sufficient. It may be augmented with “same author” or “same period” relations.

If the book store has to provide recommendations on a much finer scale (for instance, on scientific text books or philosophical essays) much finer grained domain models and user models are needed. This is even more true in a domain like multimedia content (as in LinkedTV) which spreads a broad spectrum of content and of user specific interests. Both – domains and users – need a fine grained modeling and efficient methods to relate them to each other in content filtering and recommendation generation.

3.2 User Modeling Systems

One of the first attempts to user modeling have been made in the GRUNDY dialog system [Rich79] that uses user models to create book recommendations according to users personal characteristics. These characteristics are represented by linear scaled values to show their user relevance. For getting further information about one user his user model is processed by stereotypes. To minimize the number of necessary information as much as possible the user should be classified into a stereotypical user group with the most suitable attributes as fast as possible. Based on that information the GRUNDY dialog system could infer, for example, information that most of the female people within the age between 20 and 30 are interested in romance literature. Stereotypes usually form only a basis for creating individual user models, as they help to overcome initial cold start problems that occur when no sufficient amount of information about the user is available to achieve a precise adaptation effect. For a precise adjustment an individual user model is stringently required. Here we can see that a LinkedTV user model is highly necessary to describe the user profile.

In the further progress user models became more adaptable. It was possible to record command usage or data access and its usage frequency to infer that the more a user uses a command the more it would be likely that this user is going to use this command in the future. More frequently used commands are ranked higher than others. But there was no attempt to maintain long-term preferences and user characteristics. These first user models were located in specific applications [Murray87] where no other information about the user was relevant. The user models only stored application-specific information. By analyzing the application system you can see that no explicit functional component is responsible for gathering and storing user information. There is no clear distinction between user modeling purposes and components that perform other tasks. In the following years a separation was increasingly made. The first great step to generic user modeling systems which are not related or involved to any applications is the General User Modeling System (GUMS) [Finin and Drager 1986].

3.3 GUMS

The General User Modeling System is an application-independent system that can abstract from the application system. This Software allows the definition of simple stereotype hierarchies in form of a tree structure. Programmers can describe stereotype member facts and rules prescribing the systems reasoning about them. The rules can be used to derive new information, both definite and assumed, from the current information about the user. If one fact of an assigned stereotype is in conflict with an assumption about the user, this assigned stereotype will be replaced by the next higher one in the hierarchy, which does not include the troublesome fact. By using different flags the stereotype attributes can be divided into definitely true and default attributes.

[Finin and Drager 1986] present a simple architecture for a general user modeling utility which is based on the ideas of a default logic. Although GUMS was never used together with an application system, it builds the basis for later user modeling systems [Kobsa 2001a]. Early systems usually included a representation system for expressing the contents of the user model (some logic formalism, rules, attribute-value-pairs) and a reasoning mechanism for deriving assumptions about the user from existing ones and for detecting inconsistencies. Kobsa seems to be the first author that used the term “user modeling shell system” for such kinds of software tools. Shell systems come from the field of Expert systems and have similar purposes but the general underlying aims, namely software decomposition and abstraction to support modifiability and reusability, is of course much older than expert system shells.

3.4 User Modeling Shells

User modeling shell systems provide integrated techniques, which are often used by user modeling components [Pohl 1999]. After GUMS a number of shell systems were developed which comprise different representation mechanisms for user models as well as associated inference processes.

3.4.1 UMT

UMT [Brajnik, G. and Tasso, C] allows the user model developer the definition of hierarchically ordered user stereotypes, and of rules for user model inferences and contradiction detection. Information about the user that is received from the application is classified as invariable premises or assumptions. When new information is received, stereotypes may become activated and their contents (which describe the respective user subgroups) added to the user model. UMT then applies inference rules (including contradiction detection rules) to the set of premises and assumptions, and records the inferential dependencies. After the firing of all applicable inference rules and the activation of all applicable stereotypes, contradictions between assumptions are sought and various resolution strategies applied ("truth maintenance").

3.4.2 Protum

PROTUM [Vergara, H.] represents user model content as a list of constants, each with associated type (i.e., observed, derived from stereotype, default) and confidence factor. It is related to UMT except that it possesses more sophisticated stereotype retraction mechanisms than UMT.

3.4.3 TAGUS

TAGUS [Paiva, A. and Self, J.] represents assumptions about the user in first-order formulas, with meta operators expressing the different assumption types (namely users' beliefs, goals, problem solving capabilities and problem solving strategies). The system allows for the definition of a stereotype hierarchy and contains an inference mechanism, a truth maintenance system (with different strengths of endorsements for assumptions about the user), and a diagnostic subsystem including a library of misconceptions. It also supports powerful update and evaluation requests by the application, including a simulation of the user (i.e., forward-directed inferences on the basis of the user model) and the diagnosis of unexpected user behavior.

3.5 User Modeling Server

3.5.1 UM toolkit

The *UM toolkit* [Kay, J 90, Kay, J 95] is an early user modeling server for user modeling that represents assumptions about the user's knowledge, beliefs, preferences, and other user characteristics including their personal attributes like names, date of birth and arbitrary aspects like their current location in attribute-value pairs (components). It combines modeling tools that make it possible to create a simple personal user model. Users can inspect and edit their user models [Kay, J 99]. The UM's representation of user models was strongly influenced by the goal of making the user model itself accessible via a viewer system. Each component is accompanied by a list of evidence for its truth and its falsehood. The source of each piece of evidence, its type (observation, stereotype activation, rule invocation, user input) and a time stamp is also recorded. For example the "given"-, "told"- and "observed"-evidences emerge during the user-machine interaction. When a user gives his own information by the viewer system the information gets the "given"-evidence. If the system sends a proposal to the user the information will get the "told"-evidence. And if the user executes a specific command several times it gets the "observed"-evidence. The "steroetype"- and "rule"-evidences are quite similar. They are built on rules. A "steroetype"-evidence arises e.g. for a computer scientist that he has one hundred percent interest into the movie "Matrix". The only difference is that a "rule"-evidence is used for domain knowledge while a "steroetype"-evidence comes up by statistics for a user group. The generic user model simply holds the uninterpreted collection of evidence for each component. It is only at runtime that the application uses competing specialized inference processes (the so-called "resolvers") to interpret the available evidence and conclude the value of a component. Applications have to decide which resolvers to employ.

For the LinkedTV user model system it is necessary to know where the user information comes from. For the beginning we have to declare preference links at least with a status like "explicit" or "implicit" information to differ between their relevance.

3.5.2 BGP-MS

The abbreviation stands for Belief, Goal and Plan Modeling System. It is a system that makes it possible to create a user model and to infer different assumptions from known knowledge. *BGP*-MS [Kobsa, A. and Pohl, W.] allows assumptions about the user and stereotypical assumptions about user groups to be represented in a first-order predicate logic. A subset of these assumptions is stored in a terminological logic. Different assumption types, such as (nested) beliefs and goals as well as stereotypes, are represented in different partitions (general knowledge, assumptions about what the user knows or not knows) that can be hierarchically ordered to exploit inheritance of partition contents (a partition together with all its direct and indirect ancestor partitions thereby establishes a so-called *view* of the full user model). Inferences across different assumption types (i.e. partitions) can be defined in a first-order modal logic. The data is exchanged using a specific protocol and a certain syntax. There are essentially seven commands sent by the application to the system or by the system to the application. Thus, the application notifies the system, for example, observed user's knowledge or asks what assumptions the system has about the user. Normally the data exchange between application and system is based on requests. However, it is at times useful and appropriate, to communicate certain things without being asked. For example, in case of a contradiction in statements or when a stereotype has changed. The BGP-MS system can be used as a network server with multi-user and multi-application capabilities.

3.5.3 DOPPELGÄNGER

DOPPELGÄNGER is a generic user modeling system that gathers data about users, performs

Inferences upon the data and makes the resulting information available to applications, see [Orwant, 1994]. *DOPPELGÄNGER* [Orwant, J] accepts information about the user from hardware and software sensors. Techniques for generalizing and extrapolating data from the sensors (such as beta distributions, linear prediction, Markov models) are put at the disposal of user model developers. Unsupervised clustering [Mobasher, B] is available for collecting individual user models into so-called ‘communities’ whose information serves the purpose of stereotypes. In contrast to all other user modeling shell systems, membership in a stereotype is probabilistic rather than definite. The different representations of *DOPPELGÄNGER* are quite heterogeneous. Users can inspect and edit their user models.

The main implemented application in the *DOPPELGÄNGER* project was to create newspapers that were customized on the basis of the user’s news preferences and the user’s news interests, see [Orwant, 1996]. The computations take place at spatially distributed locations and make use of portable user models that are carried by the users. The focus is set on heterogeneous learning techniques that were developed for an application-independent, sensor-independent environment. The user models were stored in a centralized database in LISP-like lists, either on fixed hard disks or on PCMCIA cards, to have the possibility to remove them for privacy reasons physically from the server. Communication between the user and the server occurs by the *“pleasant path of e-mail”*.

DOPPELGÄNGER already entered the world of ubiquitous computing since it tracked the user’s location on the basis of *active badges* and *smart chairs*: an infrared sensor at this smart chair notified the user’s workstation, when he was sitting in front of it. The centralization of *DOPPELGÄNGER’s* architecture was good for constructing a common store of personal information between applications, but didn’t scale well to modeling several people. In [Orwant, 1995] it is already concluded, that there is a need for distributed servers, the so-called DOPPELGÄNGERS.

3.5.4 CUMULATE

CUMULATE [Brusilovsky, P] is designed to provide user modeling functionality to a student adaptive educational system (see chapters 1 and 22 of this book [Brusilovsky, P. and Millán, E, Henze, N.]). It collects evidence (events) about a student’s learning from multiple servers that interact with the student. It stores students' activities and infers their learning characteristics, which form the basis for individual adaptation to them. In this vein, external and internal inference agents process the flow of events and update the values in the inference model of the server. Each inference agent is responsible for maintaining a specific property in the inference model, such as the current motivation level of the student or the student's current level of knowledge for each course topic. Brusilovsky et al. [Brusilovsky, P., Sosnovsky, S., and Yudelson, M] describe the interaction of CUMULATE with an ontology server, which stores the ontological structures of the taught domain and provides the platform for the exchange between different user model servers of higher-level information about students' knowledge.

3.5.5 Personis

Personis [Kay, J., Kummerfeld, B., and Lauder, P; Kay02] and a simplified version of it, PersonisLite [Carmichael, D. J., Kay, J., and Kummerfeld, B], have the same representational foundations as their predecessor *UM toolkit* that was still described. The components from *UM* form objects in Personis that reside in an object layer over Berkeley DB, a near-relational database system. The object database structures user models into hierarchically ordered contexts similar to the partitions of BGP-MS. It also holds objects defining the views that include components from all levels of the user model context hierarchy. The authors distinguish two basic operations upon this representation: accretion, which involves the collection of uninterpreted evidence about the user, and resolution, the interpretation of the current collection of evidence (cf. the resolvers in *UM toolkit*).

The goal of the PERSONIS project, that is based on the *UM toolkit*, is to explore ways to support powerful and flexible user modeling and - at the same time - to design it to be able to support user scrutiny and control, see [Kay, J., Kummerfeld, B., and Lauder, P]. It is novel in its design being explicitly focussed on user control and scrutability. A Personis server can support the reuse of user models over a series of adaptive hypertext systems. The so-called *views* are the conceptual high-level elements that are shared between the server and each application. The name stems from the aspect, that user model consumers might need just a few components of the user model, and a database view is applied to the whole model. The underlying representation of the user model collects *evidence* for each *component* of the user model.

The user model information in Personis is held together with the access control information as a relational object database. Several user interfaces have been implemented. The Personis *resolvers* follow the tradition of the UM toolkit and interpret the evidences at runtime. Furthermore, in [Kay, 1995] it has been demonstrated that many users can and do scrutinize their user models.

3.5.6 Deep Map User Modeling System

In [Fink and Kobsa, 2002] the user modeling system of DEEPMAP is described as the state of the art user modeling system for personalized city tours. This work was carried out in the context of the Deep Map project [Malaka and Zipf, 2000] of the European Laboratory in Heidelberg, Germany.

The DEEPMAP user modeling server is aimed to provide information in a personalized manner in the travel and tourism domain. Especially the users’ interests and preferences are taken into account. The all-in-one system offers services to personalized systems with regard to the analysis of user actions, the representation of the assumptions about the user, as well as the inference of additional assumptions, based on domain knowledge and characteristics of similar users. The system is open and can be accessed by clients that need personalization services via different interfaces. That is how the DEEPMAP user modeling system stands in tradition with [Fink and Kobsa, 2000]'s generic architecture for the user modeling shell system BGP-MS4 which proposes an object-oriented approach to realize a network-oriented user modeling shell system and which allows a distributed use of a central user model. In [Fink, 2004] the pros and cons of *directories* and *databases* are evaluated while LDAP5-based systems are recommended as a basis for user model servers. Such directories are specialized database management systems that maintain information about relevant characteristics of users, devices and services on a network. One advantage is for example the usage of predefined user-related information types. A second advantage of LDAP directories it that they can manage information that is dispersed across a network of several servers.

The DEEP MAP user modeling system provides three user modeling components: The *User Learning Component*, which learns user interests and preferences from usage data, and updates individual user models. The *Mentor Learning Component* that predicts missing values in individual user models from models of similar users, and the *Domain Inference Component* that infers interests and preferences in individual user models by applying domain inferences to assumptions that were explicitly provided by users or the two other components. The user modeling server is designed on top of an existing LDAP server as an application that is built of loosely cooperating autonomous components showing the general user modeling server according to [Fink, 2004]. The *Directory Component* consists of the three sub-systems *Communication* that is responsible for managing communication with external and internal clients, *Representation* that is in charge of managing directory content and *Scheduler* that has to mediate between the different sub-systems and components of the user modeling server. *User Modeling Components* are shown that perform dedicated user modeling tasks like collaborative filtering. The components can be flexibly distributed across a network of computers. The directory component communicates with the user model components, the clients and the models via *CORBA* and *LDAP*. There are several models that constitute the representational basis of the DEEPMAP user modeling service, namely the *User Model*, *Usage Model*, *System Model*, and *Service Model*.

DEEPMAP is part of a family of long-term research projects. One central aim of the project [Kray, 2003] was the provision of personalized tour recommendations for the city of Heidelberg that cater to an individual user's interests and preferences. The WEBGUIDE sub-project identified geographical points of interest and computed tours that connect these points via presumably interesting routes, based on geographical information, information about selected means of transport, the user model, and by the user specified tour restrictions like distance and duration. Finally such tour recommendations were presented to the user.

3.6 NoTube

The NoTube project [NoTube] can be seen in some sense as a predecessor to our LinkedTV project. It aimed at helping people to find and to choose what TV content to watch in a continuously growing amount of TV content and channels. It experimented with novel semantics-based approaches for recommending TV programs. In particular, NoTube started to use the newly evolving Web of Data (Linked Open Data, LOD) as background knowledge in order to make relations between media content items and between media content and users more specific and meaningful. It used the semantics-based approaches in order to find recommendation results for any user and in any context. NoTube has a strong focus on concrete entities (like actors, film directors, authors, movies, etc.) and their relations. NoTube summarized such Linked Open Data sets by treating them as sets of connected knowledge patterns, in order to identify their core knowledge components. Next to these semantic pattern-based approaches, the recommendation service also uses standard statistical techniques for collaborative filtering based on rating data.

3.7 Conclusion

The history of user models and their usage in content filtering, recommendation and guidance shows clearly the relation between the kind and structure of the domain and the kind of user support. The more complex the domain and the more different the users' preferences, skills, and interests are the more fine grained has the user model to be and the more computational efforts are needed to relate this fine grained user model to the content.

The Web of data with its many Linked Open Data sets provides a pool of information usable for content enrichment as well as for user models [Andrej06]. On the other side, the semantic precision and coherence within this LOD cloud is currently remarkably low. Using it as basis for user models results in a couple of completely new research challenges.

The computations used in recent decades to relate user models to domain models were quite different – from simple mappings over statistical correlations and clustering up to taxonomic and modal reasoning. What is computationally manageable in focused domains becomes unpractical if the domains are large and diverse. It is necessary to find the right trade-off between semantic granularity and precision of user and domain models on one side and the computational efforts needed to use these models for content filtering and user support.

4 The LinkedTV User Model Ontology

In Deliverable D4.2 we provided a detailed description of our user modelling approach. The main issue behind our approach is the widespread and rich domain structure typically encountered in multimedia applications, and the resulting diverse user interests and preferences. The LinkedTV User Model Ontology LUMO described in D4.2 allows us to express user interests for a broad spectrum of issues in a broader or fine grained way – depending on the specific interests of the user at hand.

In its core, a user model is a *weighted light weight ontology* collecting those concepts and concrete entities a user has a special interest in and their weights indicating the degree of interest a user has in these items.

In order to keep the content filtering on large user models and complex domain ontologies manageable we deliberately restrict the expressiveness of the user model ontology currently to:

- type relations: assigning concepts to concrete instances;
- subclassOf relations between concepts; and
- relatedTo: a selection of domain specific relations between concrete instances.

These expressive means[2] allow us on one side to represent large domain ontologies (as may be used for mental models of users in a broad multimedia domain), and on the other side to keep things relatively simple as adequate for the current state of the art in LOD ontologies. The LinkedTV semantic content filtering LSF introduced in Chapters 5 and 6 uses a simple semantic matching procedure on such light weight ontologies to compute the matching degree between a user model and a multimedia annotation (from videos, scenes, single shots). We just use the semantic relationships type, subclassOf, and relatedTo between media annotations and user models for this matching.

This ontology represents the mental model of the user. It contains the main categories and concrete instances this user is interested in. It may cover "everything in the world" – from concrete people and events up to abstract topics like certain fields in physics, politics, sports or history. We do not make any specific assumptions about how this ontology has to look like – as long as it fulfils this condition.

In D4.2 we introduced the LinkedTV user model ontology (LUMO). It contains all those concepts, topics, and concrete entities a user may be interested in. This ontology is related

[2] The relatedTo relations are an abstraction of LOD domain specific relations. In later versions we may use more specific relations with associated weights. - Currently we do not provide any means to express a user's "dislikes". The most we can say is that a user is *not* interested in something. The weight of this "something" is set to zero. The meaning of a dislike statement needs further clarification: what if a video is annotated with something the user really likes and something he dislikes? How to relate these two aspects to each other to compute the matching degree?

to various Web ontologies used for multimedia annotations. If an annotation element (represented by a URI) has to be matched with a user model element (also a URI) this match is considerably simpler if both URI belong to the *same* ontology. Otherwise general ontology alignment techniques are needed which mean more computational effort and less semantic precision (see chapter 10).

The other main aspect to be considered when using semantic user models is the way they are matched to multimedia content. In the state of the art chapter we showed that a broad spectrum of inference techniques have been applied for this task – from simple matching over statistical correlations and clustering up to modal reasoning. It is important to find an acceptable trade-off between needed inferential services and necessary computational efforts related with it.

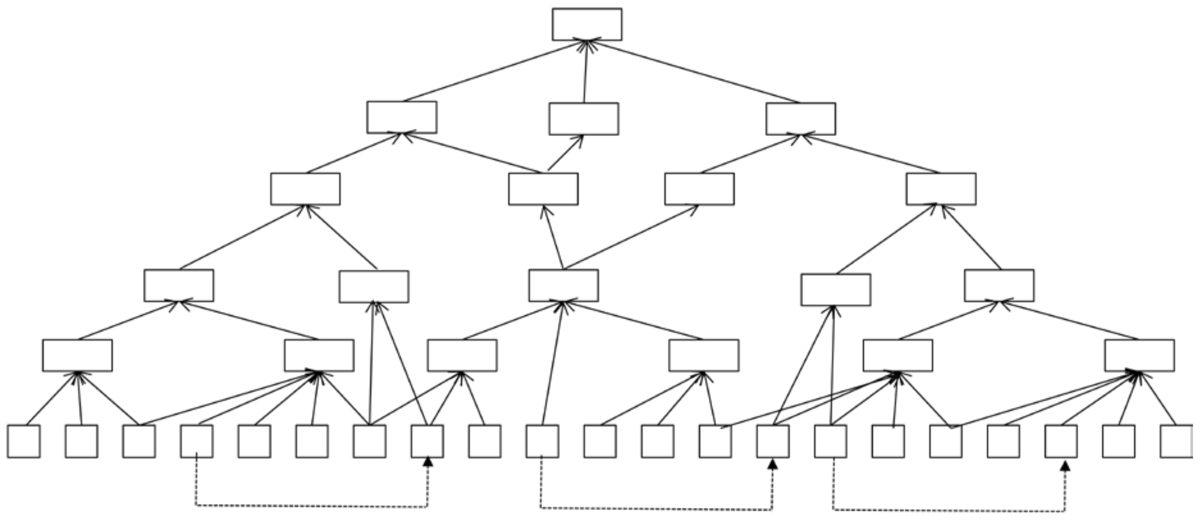
The user models and the multimedia annotations with their related Web ontologies result in quite complex models. Reasoning on them may easily become complex, too. In the following we describe a relatively simple matching algorithm which allows us to relate efficiently user models to multimedia annotations based on Web ontologies.

5 Using the Ontologies

5.1 The basic requirements for an ontology

A LinkedTV user model ontology (LUMO) consists of concepts and of instances of these concepts. Instances are assigned to their classes through 'type' relations. Classes are assigned to their superclasses through subclassOf relations. Instances as well as classes may have more than one class or superclass, respectively. Topics are a special kind of concepts which do not have instances but are connected through subclassOf relations, too[3].

Instances may be related to other instances or topics through arbitrary semantic relations (like belongsTo or controls or livesIn). In order to avoid too complicated user models we restrict our ontology at the moment to just one kind of such relations: relatedTo. A person may be relatedTo a city (living there or working there), an event may be related to a political or sports topic.



typeOf, subclassOf, subtopicOf relations

relatedTo relations (connecting instances with instances or with topics)

Figure 2: An ontology as basis of a user model

With the relations mentioned the ontology forms a directed acyclic graph.

[3] In order to keep things simple the sameAs relation is not considered as part of the ontology. We treat sameAs relations as *semantic mappings* from external terms to elements of our ontology.

We assume that all relations between ontology elements are fixed. We do not need any reasoning on such relations at run time.

5.2 Ontology modelling

In order to keep the filtering process efficient we restrict the user model ontology to those elements in which the user has some special interest and to its super concepts. If somebody has no interest in physics it does not help to have such exotic topics like Quantum Chromo Dynamics or super symmetries in this user model.

How weights of ontology elements are related to each other is a matter of experiments. Let's assume a user is interested in modern basic physics topics like black holes, Higgs particles, and dark energy. These topics will be part of his user model, and the weights are given accordingly. At the same time, the more general physics topics like cosmology or elementary particle physics will also be contained in the user model ontology in order to keep it consistent – but with different weights. And the topic of general physics, of course, will also be in the UMO as well as the natural science topic as its generalization.

We have two options how to assign weights to these more general topics:

- We treat them completely independent from the weights of the more special topics. Somebody may have a strong interest in Higgs particles but is not interested at all in general elementary particle physics. Every multimedia item annotated with 'Higgs particles' will get high ranking whereas something annotated with a related topic (like Neutrino physics) may not get any attention.
- We assume that the semantic relationship between a more special and a more general topic or concept result in a certain relationship of the user's interest in these related items. If the user did not explicitly specify an interest in the more general topics we assume an interest which is related to the one in the more special topics (reduced by a factor of 3 or something similar).

Of course, the user has to have the ability to specify explicitly his interest in every element in the user model ontology.

5.3 Multi-lingual ontologies

Ontologies can have the advantage of being multi-lingual. The concepts and instances in it have different facets for different languages. We may use such an ontology as basis for our user models. The meaning of the ontology elements is, of course, independent from natural languages. Nevertheless, for user interactions the natural language facets play an important role.

The user model ontology may be different for different users or user groups. Every user model may be based on an ontology in the user's native language. Frequently, this may result in a more natural representation of the user's model. The semantic differences between ontologies in different languages do not result from the different vocabularies used for the same things, but from the different conceptualizations used in different cultures. Of course, if a user is interested in media items annotated in a different language we have the additional mapping problem from this annotation to the user's ontology.

In the first version of content filtering we do not take multi-lingual aspects into account. The content filtering is based on the video fragment annotations represented by URI and by a user model ontology also represented as a set of URIs and relations between them.

6 The User Model

In Deliverable D4.2 we showed how the user model ontology UMO should look like in order to be used as basis of a user model. Here we show how we come from the ontology to the user model.

6.1 Weights

A user model UM is based on a user model ontology UMO (a set of URI) and weights w assigned to each element (URI) in it:

$$UM = \{w(URI) \mid URI \in UMO\}$$

These weights express the degree of interest a user has in a certain topic, element, etc.

The weights we are using are real numbers[4]: w: UMO $\rightarrow \mathcal{R}$

How weights of instances and concepts in an ontology are related to each other is nothing one can deduce from first principles. What if a user expresses strong interest in a certain topic (say Higgs bosons) but no interest at all in any of its more general topics (like elementary particle physics or general physics)? Should we deduce at least “some interest” for this user in these more general topics, or should we focus solely on that topic he explicitly expressed his interest in? In reality, things will be only rarely so rigorous, but for clarification of the basics of our approach it is necessary to answer this question. Future experiments will show what the best way is.

In the following, we adopt a non-monotonic weighting approach. The weights of a concept are inherited to all its subconcepts and instances as long as none of them has a weight on its own. If this is the case this own weight overwrites the inherited value (and is inherited to its subconcepts and instances).

If a concept or instance has more than one superconcept the strongest weight is taken.

The top level concept (“thing”) is set to weight 0. As a consequence, all items a user has no interest in inherit this weight.

6.2 Weight assignment

There are different ways to assign weights to terms in the ontology:

[4] The details are still open and may need some experiments. We can restrict weights to the interval [0, 1], we may allow any positive real number, or we may even permit negative interests to express dislike attitudes.

- Explicit user information:
  the user can express his interest in a concept or instance explicitly by assigning a weight to it.
- Preference learning:
  the system observes the user's media consumption and increases the weights of those ontology terms which are related to the consumed videos in an appropriate way[5].
- The weights in the user model may be decreased if the user did not show interest in related media items for a certain amount of time.

These aspects are described in more detail in D4.2. At the moment we simply assume that all weights are given – without any specification where they came from.

6.3 Contextualised User Model

The contextualised user model (CUM) is based on a part (subgraph) of the user model UM. Some elements in the ontology are selected as 'active', and all its descendants (subconcepts, instances, relatedTo[6] terms) are active, too.

$$CUM = \{w(URI) \mid URI \in UM \wedge state(URI)=active\}$$

We assume that the weights of the active ontology elements forming the CUM are the same as in the full user model. The weights of those ontology terms not active in the CUM are effectively set to 0 as long as the user is in the context represented by this CUM.

6.4 Annotated videos and reference ontologies

In D2.2 an annotation VA for a video or any of its fragments (scenes, shots) is defined as a set of *ontology terms* (instances, topics, concepts) represented as URI:

$$VA = \{x \mid x \in O\}$$

where O is a Web ontology like DBPedia, Schema.org. MusicOntology, etc. We call this Web ontology O the reference ontology of the annotation element x.

Each annotation element x may have a video weight v(x) (a real number) representing the relative importance this element has in the video (or in a scene or frame within this video). This weight v(x) is user independent and *not* related to the user model weights w.

Because the annotations are elements of a Web ontology O we need mappings from these annotation elements to terms in our LinkedTV user model ontology LUMO.

We have two different situations:

[5] The details still have to be specified.

[6] The treatment of relatedTo is not clear at the moment: if for instance 'politics' is activated in the CUM a politician (which is neither a subclass nor an instance of politics) should be considered, too.

1. The reference ontology O or those parts of it which contain the annotation element x are directly related to elements in the user model ontology UMO through equivalent class or sameAs relations. Then we map the annotation element x to that element y in the UMO which is most specific in LUMO and connected though am equivalent class or sameAs relation to x’s reference ontology O.
2. The reference ontology O is not directly related to LUMO. Then we need a heuristic mapping which approximates the non-existing explicit sameAs relation through a “most similar” relation to a term y in LUMO. We use the name similarity as criterion. More sophisticated matching procedures will be applied in forthcoming versions of content filtering.

7 Content Filtering

In the following we outline how the contextualised user model CUM and the multimedia fragment annotation VA can be used for content filtering[7]. The criteria chosen to specify this approach can be summarized as follows:

The user model UM as well as the multimedia fragment annotations VA are based on light weight ontologies available in the LOD cloud on the Web. This has the advantage that a rich repertoire of meaningful information can be used but at the same time we inherit all the problems associated with pure semantics, unclear relationships, and inconsistencies in this LOD cloud.

The multimedia Web is already today a large repository of content and it will grow considerably in the future. Consequently, we need efficient filtering procedures in order to manage this large set of data. Though powerful reasoning techniques exist which may be useful in content filtering we decided to focus onto an efficient matching procedure. It is adequate for the light weight ontologies underlying our user models and our multimedia fragment annotations. In chapter 10 we give introduce a more powerful reasoning tool which can improve content filtering in combination with the semantic matching approach introduced in this chapter.

Semantic filtering can be used for a couple of services:

- Every scene or frame within a video is annotated with a set of terms (URI). The interest a user has in them may vary considerably. A user model can be used to rank these annotations according to the special interest of the user – avoiding information overload.
- A user may point to an object of his interest in a video scene. The simplest way is just to show the URI associated with this object to the user (in an appropriate form). But typically there are many different kinds of information available related to this object, and the user model may be used to highlight that information which is of special interest to the user in this context[8].
- When a user watches a video he may be interested in *related* content. On one side, we may use a user independent similarity approach. In this case the ontology gives us a semantic similarity measure.

---

[7] The content filtering approach introduced here is an extension and modification of the “semantic finger print” matching developed in a previous project by our partner Condat AG [Alia06, Geihs09].

[8] In our RBB use case we may have a report about an event in a village. The church in this village appears in one scene. If the user points on this church he may be interested in many different things: the history of the building, some religious issues, the architecture, etc. The user model may be used to rank those aspects and to highlight those to the user which typically are of special interest to him in a movie pointing to one of the characters in a scene may be more dedicated to the character himself or to the actor playing this role, etc.

- Alternatively, we can use the user model to find out those videos which are similar "in the eyes of the user". The user model may be used to filter this similarity according to the user's preferences.

These services are described in detail in the following:

7.1 Selecting preferred links

Let's assume an annotation VA = $\{x_1, x_2, \ldots, x_n\}$ with x_i being URI for objects shown in this scene or frame the user is just watching. Typically, there are too many annotation elements to be presented to the user in an unfiltered way. Thus we want to use the contextualized user model CUM to rank this annotation set.

First, we map the annotation elements x_i in VA to the related elements y_i in the user model ontology UMO (see chapter 4). We call this annotation set the user-specific annotation set UVA:

$$UVA = = \{y_i \mid x_i \in VA \ \wedge \ mappedTo(x_i, y_i) \wedge y_i \in UMO\}$$

Then the filtering can be based on two aspects:

- The weight $v(x_i)$ the annotation elements have in the annotation VA.
- The weight $w(y_i)$ the mapped annotation element has in the contextualized user model CUM.

The filtering procedure just takes the product

$$v(x_i) * w(y_i)$$

of these two weights and ranks the annotation elements accordingly.

The selected annotation elements are elements in the ontology O. Alternatively, the user may be interested in other links (URIs) dealing with the same object. A semantic mapping is needed allowing us to relate the ontology elements to related information outside our ontology.

7.2 Extending preferred links

The simple filtering approach in section 6.1 just takes the links contained in the annotation set VA. Sometimes, a user may be more interested in things somehow *related* to these annotation elements. For this purpose, we extend the original annotation VA to an extended video annotation EVA:

$$EVA = VA \cup \{z \mid relatedTo(x, z) \wedge x \in VA\}$$

The video weight v(z) assigned to the related ontology terms $z \in EVA \setminus VA$ is the same as the one of the original annotation element[9] $x \in VA$:

$$v(z) = v(x) \text{ where relatedTo}(x, z)$$

Alternatively, we may introduce a weight factor ρ for relatedTo expressions. In this case all relatedTo elements get a video weight reduced by this factor compared with the original element:

$$v(z) = \rho * v(x) \text{ where relatedTo}(x, z)$$

Now, the filtering procedure can be applied to the elements in EVA in the same way as for the VA.

A critical point is the number and kind of relatedTo links provided in LOD ontologies. DBPedia, for instance, comes with a large set of relations for many entities - some of them with clear semantic meaning, others of quite low value from a semantic modeling perspective. We have to find a way how to differentiate useful relations from the other ones. In a first attempt we can manually select those relatedTo link categories we want to use for our purposes.

It will be a matter of user interaction capabilities how a user can select between these two options. It is also a matter of user interaction capabilities how to present the filtered and ranked links. We may restrict the list presented to the user to a fixed length, we may highlight the first N links but allowing the user to scroll down to the less ranked ones, etc.

7.3 Filtering related links for a video object

Instead of using the whole annotation set VA in a given scene or frame the user may point directly to one of the objects x in this scene. Now the same procedure as outlined in chapter 6.2 for the EVA can be applied to this one annotation element and its related terms. Because this set is (typically) much smaller than in the previous case much more information can be provided to the user fitting into this focus.

7.4 Semantic Similarity of two videos

Following Condat's Semantic Finger Prints approach [Alia06, Geihs09] we can define the semantic similarity of two multimedia fragments. The semantics is defined in the reference ontology O – without any relation to a user model. In the following chapter we will provide a personalized version of semantic similarity.

[9] If a related element y is related through more than one annotation element $x_1, x_2, \ldots, x_n$ the maximum weight of these elements is taken: $v(z) = \max(\{v(x_1), v(x_2,), \ldots, v(x_n)\})$.

Later we may use *different* semantic relations with different weights allowing us to better differentiate the user model.

The similarity of two media fragments is defined according to their extended annotation sets EVA1 und EVA2 and the reference ontology O defining semantic similarity of the elements in these sets[10].

All elements x in the EVAs are weighted with their video weights v(x).

In order to determine the semantic similarity of EVA1 and EVA2 w.r.t. the reference ontology O we have two tasks to fulfill:

- pairing: find out which annotation elements on each side are similar to each other; and
- what is the semantic similarity of each pair.

Pairing is easy if both EVAs contain the same element x: $x \in$ EVA1 and $x \in$ EVA2.

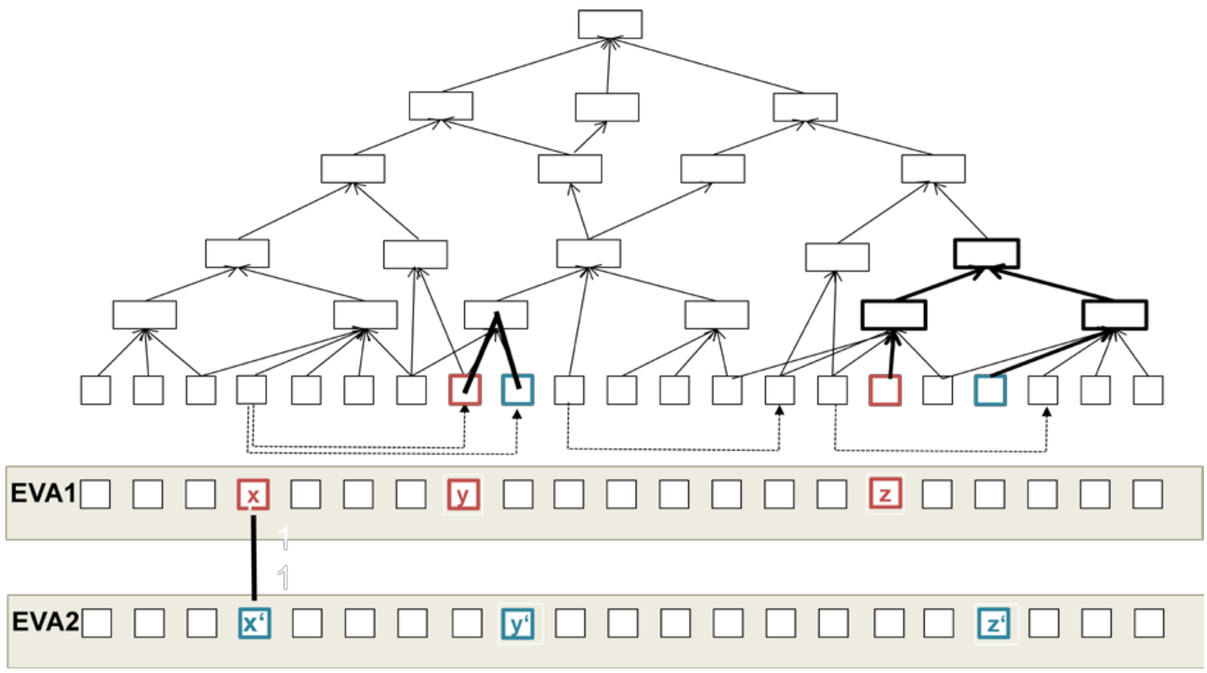


Figure 3: Semantic similarity of two videos

The contribution δ of the pair <x, x> is the product of x's weigths in the two annotations:

$$\delta(x, x) = v_1(x) * v_2(x)$$

if for an element x1 in EVA1 there is no direct correspondence in EVA2 we have to find the semantically most similar element y w.r.t. the ontology O. As measure of the semantic similarity we take the path length in the ontology from x to y (see Figure 3):

- if both x and y belong to the same most specific class c the path length λ is 2;

[10] Here we assume that both videos are annotated with elements from the same reference ontology. If this is not the case we have to apply semantic mappings between them.

- if they both belong to the first superclass of their most specific classes the path length λ is 4; etc.

For every element x in EVA1 we assign the most similar element y from EVA2.

The longer the path is between x and y the less is the similarity between them. We may assign a constant factor n to each step in the path so that the overall weight reduction from a path with length λ is n^{λ} .

Consequently, the contribution of a pair x and y of annotation elements to the semantic similarity is

$$\delta(x, y) = v_1(x) * v_2(y) * n^{\lambda}$$

The overall semantic similarity σ of two videos represented by their extended annotation sets EVA1 und EVA2 is

$$\sigma\,(EVA1, EVA2) = 1/N * \Sigma\ \delta(x, y) \text{ with } x \in EVA1 \text{ und } y \in EVA2$$

and N is a normalization factor based on the video weights of all annotation elements.

This semantic similarity approach is completely based on the *structure* of the ontology O. It does not take any weights of concepts, instances, or relations into account. Alternatively, we may assign higher weights to concepts which are sparsely populated, to instances only rarely mentioned or considered as typical for a certain concept, etc. This will be a matter of experimentation and discussion.

7.5 User specific semantic similarity of annotated multimedia fragments

The semantic similarity approach described in chapter 7.4 is user independent. A user may have a personal view on similarity based on his different interests in various areas. A video on Mozart and Salzburg may cause different associations for a music fan and somebody interested in beautiful cities. Consequently, the music fan would consider a video on Haydn much more similar to the first video than a video on Basel – which the city lover would consider more relevant to him.

The approach described in chapter 7.4 can straightforwardly be extended for this purpose: instead of just considering path lengths as basis for semantic similarity we take the weights assigned to nodes on this path into account. We have different options:

- the weight of the top level node: this node characterises the common view this user has on the two related annotation elements x and y. Both are considered instances or subtopics of this common concept. Consequently, this weight stands for the interest the user has in these issues;
- the average weight along the path: if the user has different interests in those concepts which relate the two items the (geometric) average in their weights may be a good approximation in the user’s overall interest in these things;

- the maximum weight along the path.

Which measure produces the best results will be a matter of experimentation.

7.6 Other filtering criteria

In addition to filtering criteria which are related to the user's mental model (and the ontology representing it) we have often some other selection criteria.

Some of them are

- The kind of the video: documentary, report, drama, action, comedy, etc.
- The content level characteristics of the video: high level, introductory;
- The content quality criteria: high quality, main stream, low quality, trash; etc.
- The technical level: HD, SD, amateur, …
- Other meta data: time and place of production, duration, …

Each video may be described along these different dimensions. A user may specify his selection criteria accordingly. A match between video description and user criteria has to be integrated into the selection/filtering process. This matching has also to be based on semantic techniques enabling conditions like “most recent”, “best quality”, not shorter than 5 min. etc.

8 Use Cases – User model

This chapter describes six different examples of user characters and makes assumptions about the resulting LinkedTV user model which sets the initial state of the user model depending on the LinkedTV user model ontology for our concept and content filter algorithm. Of course, since this is the first draft of user model use cases there are still many unknowns at this stage of the project. This is not a mandatory default specification for user profiles. Many features may be considered as optional and we have to choose the most promising ones or extend the profiles with additional information, but it should be appropriate for basic user models.

At the beginning we describe some examples for typical users. The descriptions contain demographic information like age, gender or education and further more preference-based categories like favorites in different areas or hobbies. Additionally, you find some contextual information that is useful for later work. The user model does not only depend on explicit user inputs. Nearly the whole data could be extracted from social media networks. It also can infer concepts from user profiles in Facebook. But to improve the user models expressiveness and integrity explicit user inputs are reasonable. The crucial point is that automatically inferred concepts and explicit entered concepts together have to construct an approximately realistic user model with appropriate weightings.

8.1 Characters and their user models

For our demonstrations we defined – according to our use cases outlined in WP6 – six prototypical user characters. Each of them is described with their biographical and social data, their professions, interests, and hobbies.

Lukas: German single male, age: 35, technician

Sophie: Dutch student**,** age: 20, student of the lectureship of history and arts

Jose: Spanish elder, age: 62, pensioner

Ralph: Sports loving carpenter, age: 19, German

Nina: Urban mom, age: 32, teacher on maternal leave

Peter: Socially active retiree, age: 65, German iving in Potsdam/Brandenburg

Comprehensive descriptions of these user examples are given in the appendix 14.1.

The next step is to translate the verbal descriptions of user models introduced into ontology based user models to be used in the LinkedTV Semantic Filtering LSF.

We have extracted several concepts and individuals characterising the user profiles. These terms have to be part of our LinkedTV user model ontology. So, we can recognize missing concepts to refine our ontology (see fig. 4). Our idea is to offer the user the opportunity to enter explicit information in a kind of tree view. The user can choose and click different concepts from our ontology that you mostly find in e.g. social network profile management services (Facebook, Xing, etc.). The user model itself represents only specific parts of the ontology in list form. The filter algorithm can apply that user model to a concrete video and calculate the video relevance for this user.

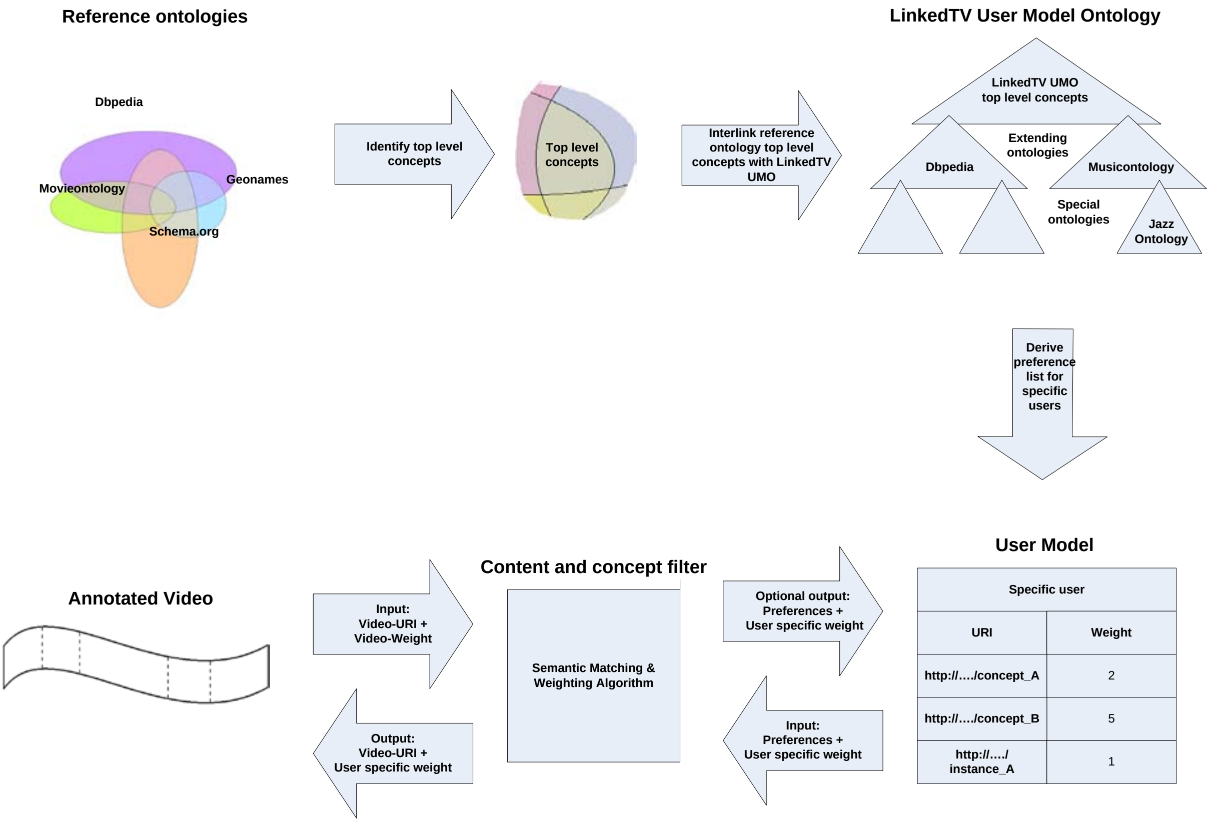


Figure 4: Conceptual workflow of creating the user model ontology up to the user model and its usage by the LSF algorithm

This cutout constitutes the input for our filter algorithm. It contains the relevant information to describe the user preferences and later we will add a new entity for contextual information which restricts the user model list to only the actually relevant preferences. Here we restrict the scenario in such a way that the complete user model can be used at any date.

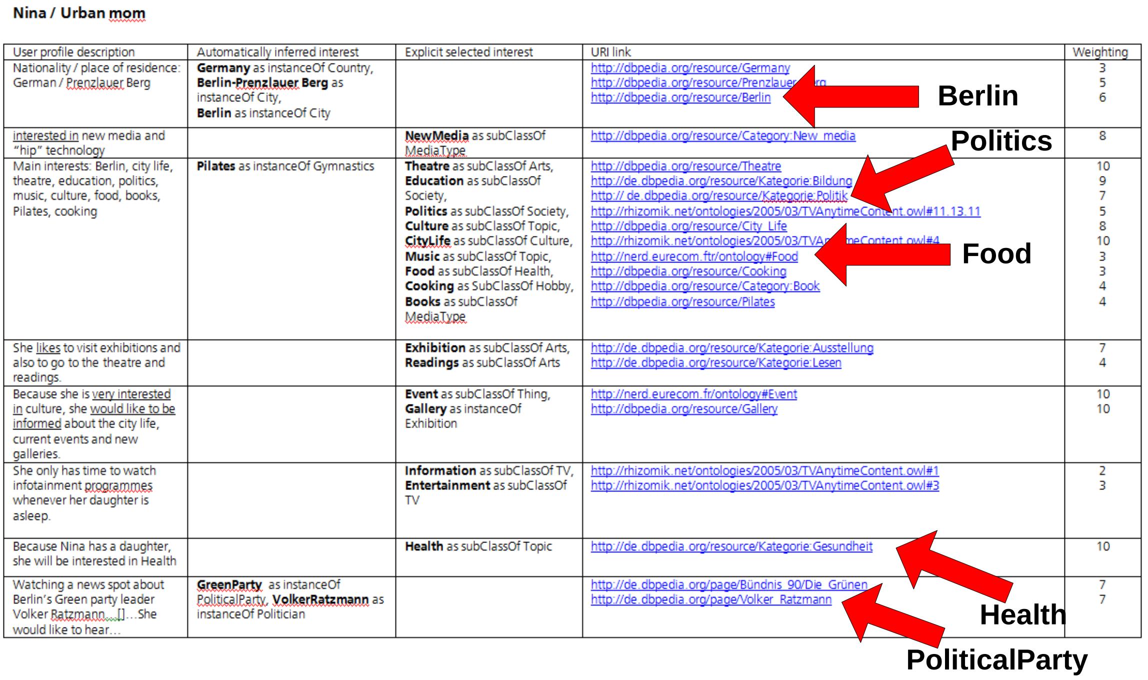
The user model examples shown above are just illustrations of how different user interests can be in a multimedia world. Everything can play a role in a user's mental model and in his preferences in media consumption. In order to relate these interests to multimedia content ontologies are needed. They provide the necessary semantic mapping from multimedia content annotations to more or less specific user interests which are sometimes expressed as abstract topics ("local politics", "health"), and sometimes as concrete entities ("Potsdam", "Mozart").

8.2 How to use user models in content filtering

In the following we illustrate how user models are used to filter annotated multimedia fragments according to a user's preferences.

First we describe (parts of) the user model of our example: Nina, the urban mom (see Appendix 14.1). In a next step we show some example shots with their annotations from a RBB use case video. Finally, we demonstrate how the user model and the multimedia fragment annotations work together in the LSF filtering procedure to highlight those items in the multimedia content which fit best to the user's preferences.

Step 1: An exemplarily user model



Nina / Urban mom

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weighting
Nationality / place of residence: German / Prenzlauer Berg	**Germany** as instanceOf Country, **Berlin-Prenzlauer Berg** as instanceOf City, **Berlin** as instanceOf City		http://dbpedia.org/resource/Germany http://dbpedia.org/resource/Prenzlauer_Berg http://dbpedia.org/resource/Berlin	3 5 6
interested in new media and "hip" technology		**NewMedia** as subClassOf MediaType	http://dbpedia.org/resource/Category:New_media	8
Main interests: Berlin, city life, theatre, education, politics, music, culture, food, books, Pilates, cooking	**Pilates** as instanceOf Gymnastics	**Theatre** as subClassOf Arts, **Education** as subClassOf Society, **Politics** as subClassOf Society, **Culture** as subClassOf Topic, **CityLife** as subClassOf Culture, **Music** as subClassOf Topic, **Food** as subClassOf Health, **Cooking** as SubClassOf Hobby, **Books** as subClassOf MediaType	http://dbpedia.org/resource/Theatre http://de.dbpedia.org/resource/Kategorie:Bildung http:// de.dbpedia.org/resource/Kategorie:Politik http://rhizomik.net/ontologies/2005/03/TVAnytimeContent.owl#11.13.11 http://dbpedia.org/resource/City_Life http://rhizomik.net/ontologies/2005/03/TVAnytimeContent.owl#4 http://nerd.eurecom.ftr/ontology#Food http://dbpedia.org/resource/Cooking http://dbpedia.org/resource/Category:Book http://dbpedia.org/resource/Pilates	10 9 7 5 8 10 3 3 4 4
She likes to visit exhibitions and also to go to the theatre and readings.		**Exhibition** as subClassOf Arts, **Readings** as subClassOf Arts	http://de.dbpedia.org/resource/Kategorie:Ausstellung http://de.dbpedia.org/resource/Kategorie:Lesen	7 4
Because she is very interested in culture, she would like to be informed about the city life, current events and new galleries.		**Event** as subClassOf Thing, **Gallery** as instanceOf Exhibition	http://nerd.eurecom.fr/ontology#Event http://dbpedia.org/resource/Gallery	10 10
She only has time to watch infotainment programmes whenever her daughter is asleep.		**Information** as subClassOf TV, **Entertainment** as subClassOf TV	http://rhizomik.net/ontologies/2005/03/TVAnytimeContent.owl#1 http://rhizomik.net/ontologies/2005/03/TVAnytimeContent.owl#3	2 3
Because Nina has a daughter, she will be interested in Health		**Health** as subClassOf Topic	http://de.dbpedia.org/resource/Kategorie:Gesundheit	10
Watching a news spot about Berlin's Green party leader Volker Ratzmann…[]…She would like to hear…	**GreenParty** as instanceOf PoliticalParty, **VolkerRatzmann** as instanceOf Politician		http://de.dbpedia.org/page/Bündnis_90/Die_Grünen http://de.dbpedia.org/page/Volker_Ratzmann	7 7

Figure 5: Parts of the user model

This figure shows a user model of Nina, an urban mum, who we have already described. As you can see, the model includes URI's with their weightings. Additionally we have highlighted some concepts that will be important in the next steps, because they occur in the video annotations of the following video.

Step 2: The appropriate concepts in the LinkedTV user model ontology

Figure 6: The LinkedTV user model ontology underlying this example

In Figure 6 we present a segment of the LinkedTV user model ontology LUMO with highlighted terms from Figure 5.

Step 3.1:

Video annotation of the RBB video:

Kontraste_20111103_lebensmittel_m_16_9_512x288.mp4

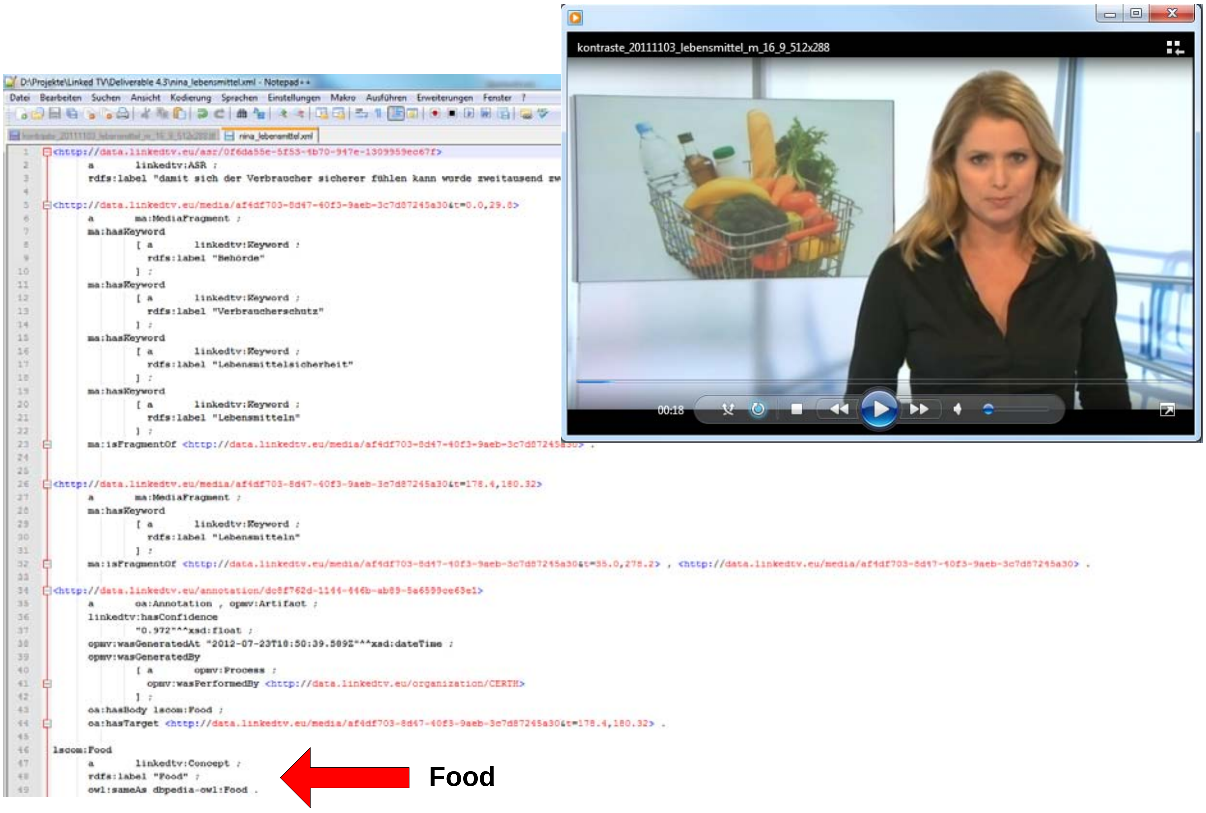


Figure 7: A video scene with annotations (1)

Step 3.2:

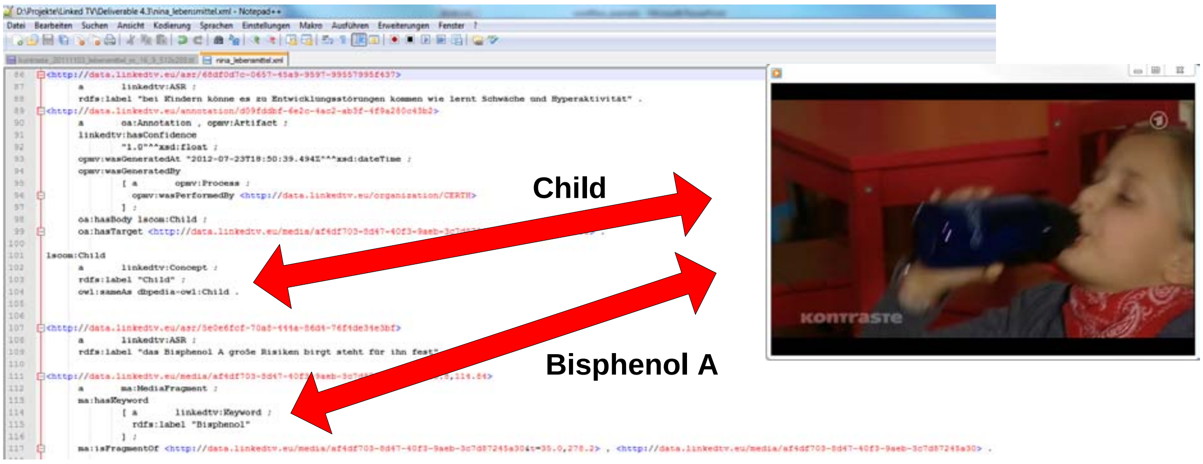


Figure 8: A second video scene with annotations (2)

Step 3.3:

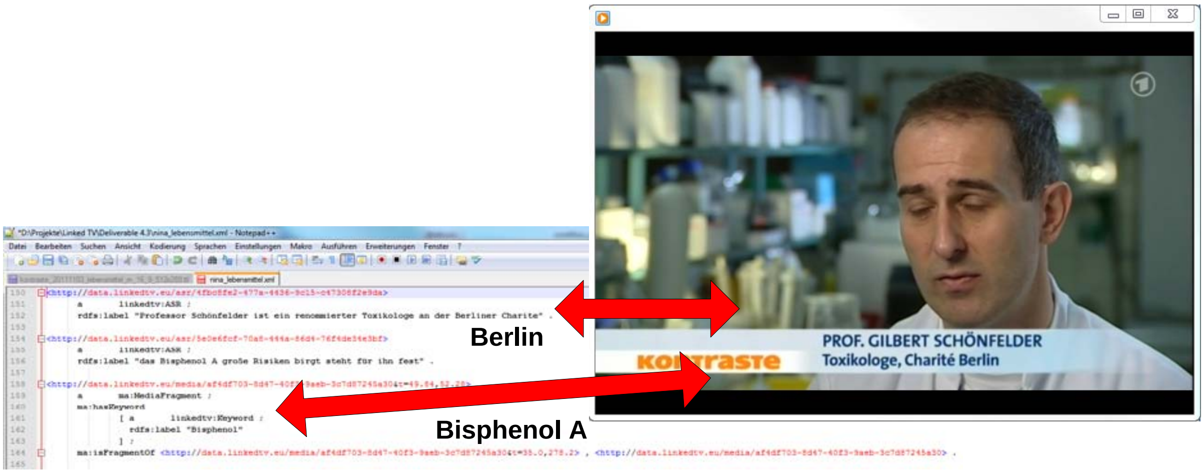


Figure 9: A third video scene with annotations (3)

Step 3.4:

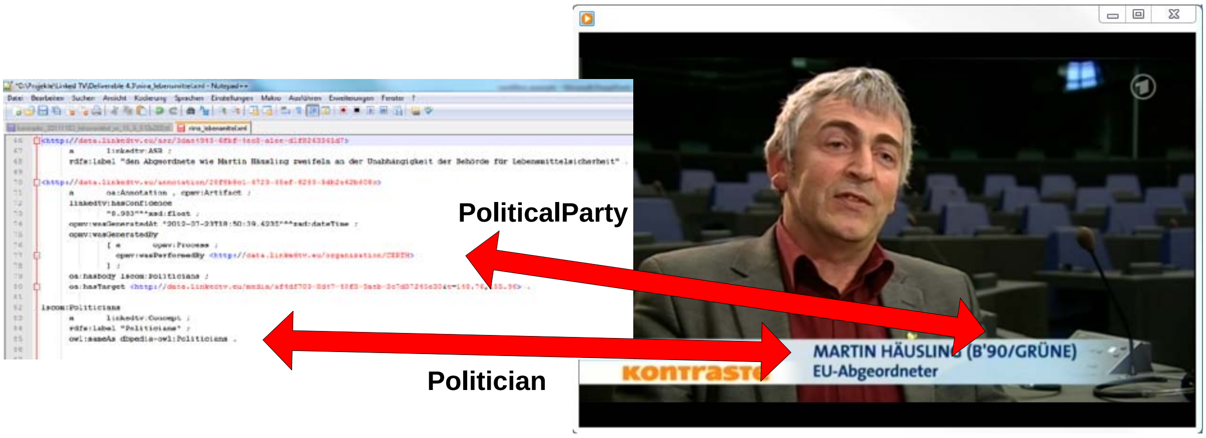


Figure 10: Fourth video scene with annotations (4)

Step 3.5:

Figure 11: Fifth video scene with annotations (5)

The five video scenes and the corresponding annotations show that each scene has its own annotations. Most of the identified concepts and instances in the scene annotations come from DBPedia. The concepts show what the scene content is about.

Step 4:

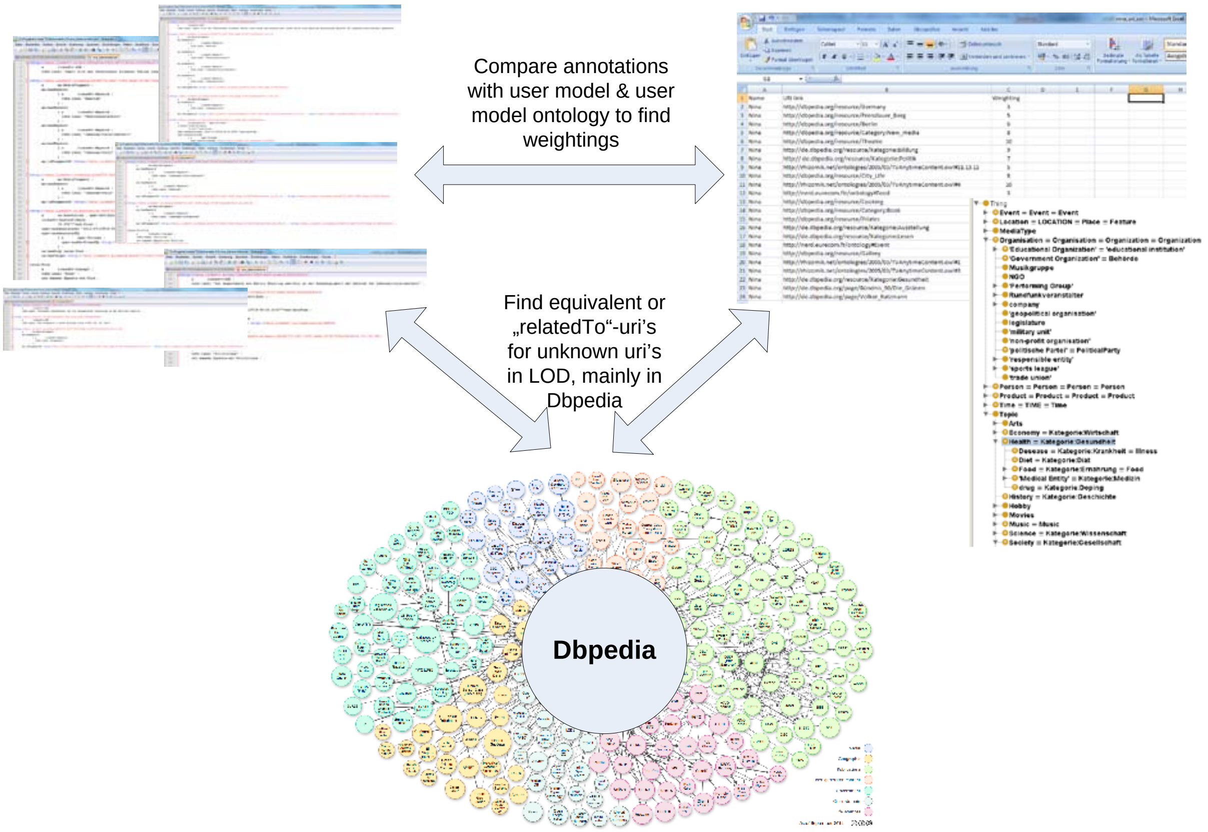


Figure 12: Relevant components for our LinkedTV Semantic Filtering (LSF) algorithm

The LinkedTV Semantic Filtering (LSF) algorithm has access to the concrete user model, the multimedia annotation file, the LinkedTV user model ontology (LUMO) and some reference ontologies in the web (e.g. DBPedia). Now LSF can compare video annotations with user model URI's to find accordances, or look for reference concepts in DBPedia which are not included in the LUMO but may have an equivalent concept or a relation to some concepts in UMO.

Step 5: Direct weighting of known terms

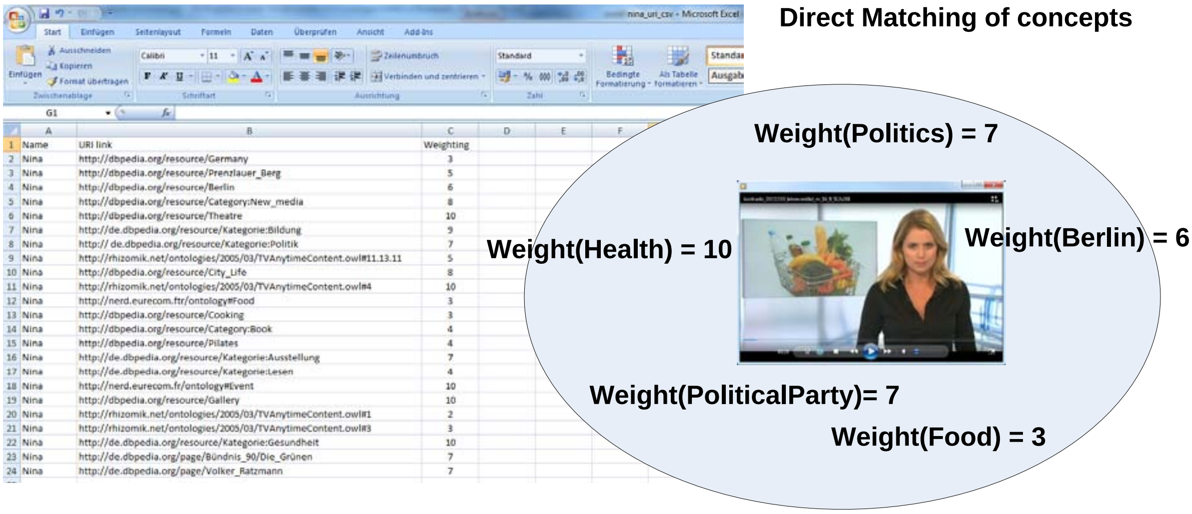


Figure 13: Direct weighting of known terms

Directly matching concepts between LUMO and the video annotation file can be easily weighted. If “Politics” in the video annotation is present in the user model – what is the case in our example – the LSF assigns the according weight of the user model URI to the video annotation URI.

Step 6: Mapping unknown terms and their relations

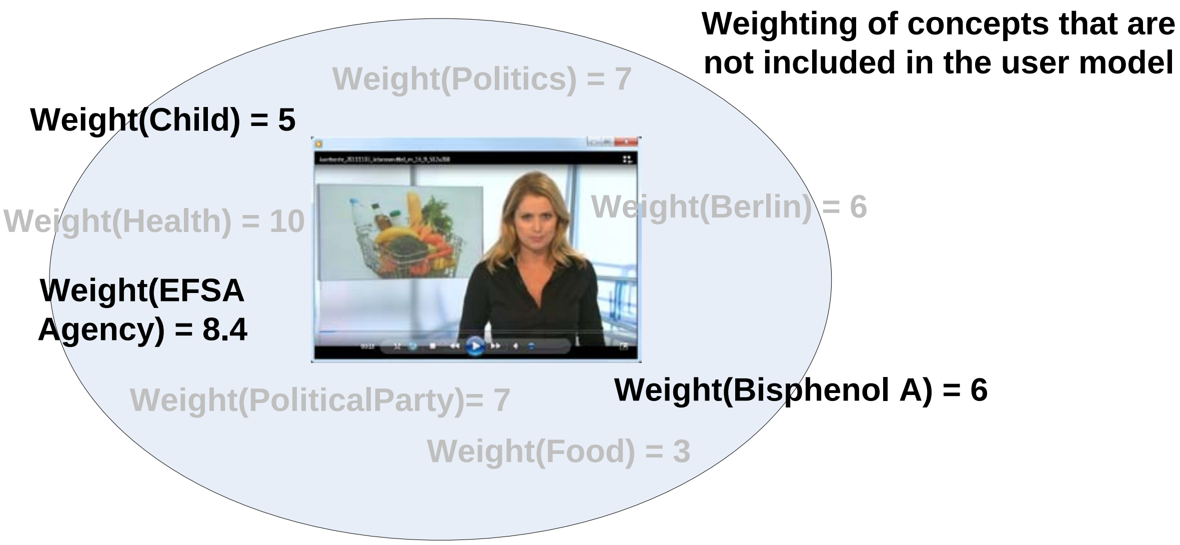


Figure 14: Mapping of unknown terms

This step describes how to handle multimedia fragment annotations which are in not contained directly in the user model. Semantic relations between these annotations and user model elements represented in LOD ontologies are used to relate these two sides:

The term BisphenolA has its own DBPedia URI. Its type is an „instanceOf“ a toxic substance. Our algorithm can’t find an equivalent concept in our LinkedTV user model ontology. So we determine the next upper class of Bisphenol A until we find a concept that matches with our ontology. In several steps we find out that Bisphenol A is subordinated to the concept “Chemistry”. This concept is contained in the LinkedTV user model ontology, but has no weight in Nina's concrete user model. In this case Bisphenol A would get the weight 0. But there is another important type of Bisphenol A. Next to the "instanceOf"-relation it also has a relatedTo-relation to Health. Since the concept "Health" is part of our ontology and Nina's user model we can inherit a weight for Bisphenol A from the concept Health. Considering the factor that has to be defined for this relation the term Bisphenol A gets a weight lower than that of Health. As you can see Health has a weight of 10, because it is very important for Nina. The relation between Bisphenol A and Health has a factor like 0.6. You get a new weight for Bisphenol A by calculating the values (10 x 0.6 = 6).

Similar to this you can compute a weight for EFSA-Agency occurring in the video. This agency is a subconcept of our concept "GovernmentalOrganization" in the LUMO. Our algorithm identified this concept by traversing the DBPedia hierarchy until a matching concept in our ontology is found. The agency is now related to our user model. It is related to the concept "Health" and "Food". LSF can calculate a derived weight by (Health(10) x relation-factor (0.6)) + (Food(3) x relation-factor(0.8)) = 6+2.4 = 8.4 for the new term "EFSA-Agency".

This process is done for any URI that is annotated with the multimedia fragment. If our algorithm LSF detects a strong or weak relation between the annotation and the user model it can assign a weight to the annotation. This has impact on the entire weight of the multimedia fragment. The directly weighted concepts and the mapped concepts are accumulated and increase the relevance of this multimedia content for this specific user.

9 Efficiency and semantic precision

Efficiency and semantic precision are relevant criteria for a filtering approach.

We do not expect that the set of concepts in the LUMO changes so frequently. Of course, every day some new terms appear in the media – but they are primarily related to instances (events, people) or to basic topics. Formal knowledge representation tells us that this may be used to form new concepts, too. The “Euro financial crisis” is a new topic and forms a subtopic under the European political topic and maybe the European financial topic (if the user’s model is so fine grained in this domain). This is something we can find out from an updated DBPedia ontology or similar Web ontologies. All political meetings related to this crisis management may form a new concept “Euro financial crisis management meeting”. We may doubt if that is a useful new concept. The combination of the new topic “current Euro financial crisis” and the existing concept “political meeting” will be sufficiently detailed to represent this context adequately. A multimedia fragment annotated with these two terms will be used in (nearly) the same way as one annotated with an instance of the new concept “Euro financial crisis management meeting”.

As a consequence, we do not have to adapt the concept hierarchy in the LUMO frequently. There will be new instances and basic topics which are assigned to existing concepts and topics and which can be integrated into the ontology straightforwardly.

Another important issue is the size of the user models. Though there are currently no empirical data about this subject we do not expect user models with more than a (few) thousand concepts. Also the set of concrete instances a user is interested in is expected to be limitied to this order.

As outlined in the previous chapter efficiency and quality of semantic filtering will be the better the better the underlying ontologies on the user side and the multimedia annotation side are aligned. This is the key issue. This includes semantic granularity issues on both sides: if a user has specific interests the multimedia annotations should be of a comparable granularity. Reasoners (like the one introduced in Chapter 10 of this document) can help to find the needed fine grained annotations or the right matches between annotations and user model elements.

These semantic granularity and data complexity issues are also related to the way the future multimedia world will be organised. Do we have a few general, global, widespread providers (like YouTube or MyVideo) for all kinds of multimedia and TV content? Or do we still have specialized channels like the German ARD focussing on German users? In the first case annotations have to be more precise w.r.t. locations and other context issues in order to be able to discriminate German football matches from Greek or Mexican ones, and to filter out German political events from those anywhere else (which in general may be not so interesting for a typical German user).

LSF relates the whole multimedia fragment annotation VA to the complete user model UM. This summary view allows us to accumulate meaning from different matches. A media fragment will typically not only mention a person or an event – it will also describe what the person is doing, what the event is about, etc. Additional information from the Web (relatedTo) allows us to enrich this annotation semantically. So, there is not only a match between a person in the multimedia annotation and a kind of people a user is interested in but also a match between related terms (like topics this person is dealing with).

10 LSF: the LinkedTV Semantic Filtering Tool

In order to demonstrate the concepts described in Chapter 7, a first version of the semantic content filtering LSF V0.1 has been implemented. This implementation is dedicated to LinkedTV system developers to analyse and evaluate the whole approach from a more technical perspective. It is not designed directly for the end user. Most software components in current implementation will be replaced in the future by components which are more intuitive for the end user. The system is still under active development and will be available for online access as soon as possible.

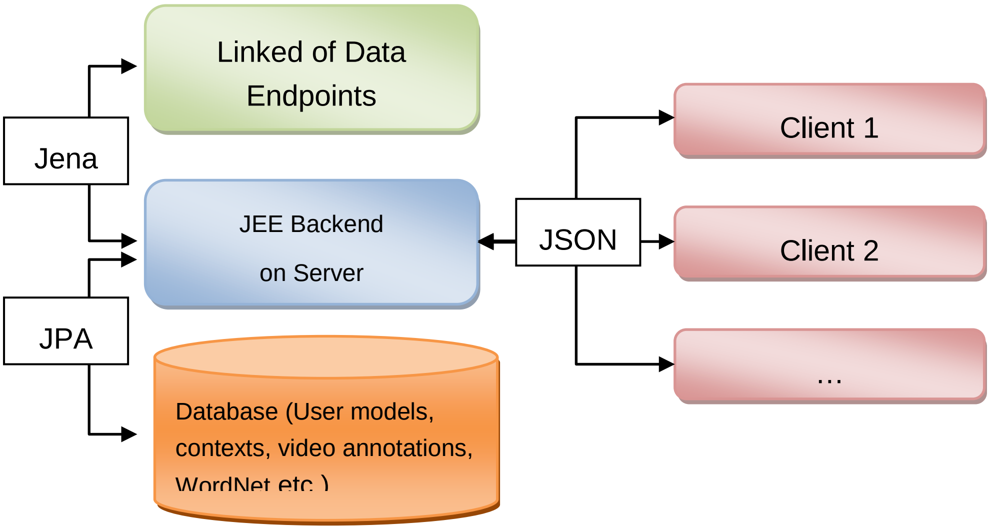


Figure 15: The current LSF system architecture

10.1 Introduction

Generally, LSF is a web-based Rich Internet Application (RIA) with JEE as server-side backend. The client side is designed with “Single Page Application” and “Responsive Design” in mind. This makes the application more flexible and versatile on different platforms (desktops, smart phones, tablets, etc.). The architecture of the overall system is illustrated in Figure 15. In general the application consists of following functional components:

- A set of user models. Each user model is represented as a list of weighted interests and each interest has a unique URI (from user model ontology or reference ontology), see Figure 16.
- A set of contextualised user models CUM for different conctexts. In Figure 17 and Figure 18 the user model of user MK for Context2 and Context6 are illustrated, respectively. Both user models are associated to the user MK, but for a given context only one user model is active.

- A set of multimedia fragments and each of them is associated with a set of weighted annotations provided by WP2 (see D2.2). A sample set of weighted multimedia fragment annotations VA is depicted in Figure 19.
- Four algorithms to correlate the contextualised user model CUM and the weighted multimedia fragment annotations VA for further video filtering. They are precise matching, relatedTo-based, WordNet-based and pattern-based matching. More about these algorithms see Chapter 10.3.
- Visualisation of the overall correlation process. The dynamic searching process for URI correlation is of great importance for validating the whole approach. Since all the complex correlation algorithms run on the sever, all necessary information are persisteent there and pushed, visualised “interactively” on the client side. A sample search-and-correlation tree is illustrated in Figure 22.

Figure 16: Four user models and user MK has four weighted interests denoted as URIs

10.2 The implementation of “relatedTo”

As described above, the abstract relation “relatedTo” is used to connect two topics or individuals with some semantic relationships. We can manually select those domain specifc relations we are treating as relatedTo.

In the current implementation we have selected manually[11] several concrete “relatedTo” relations or relations as in Table 1.

[11] This approach of selection and weighting relatedTo relations is a first attempt and needs further investigations.

Table 1: Manually specified "relatedTo" relations in current development version

relation	Factor
rdf:type	0.7
rdfs:subClassOf	0.7
dbpedia-owl:leaderName	0.5
dbpedia-owl:governmentType	0.5
dbpedia-owl:language	0.7
dbpedia-owl:largestCity	0.3
dbpedia-owl:currency	0.3
dcterms:subject	0.7
dbpedia-owl:birthDate	0.7
dbpedia-owl:birthplace	0.7
dbpedia-owl:president	0.7
dbpedia-owl:almaMater	0.3
dbpedia-owl:chancellor	0.7

To make the approach more flexible a "weight factor" has been introduced for each property (see the second column in Table 1). This provides even finer granularity and makes the correlation process more realistic (also cognitively more practical. For example, if class C is subclass of class B, class B is subclass of class A, then according to the factors in Table 1, the "relatedTo" relationship (rdfs:subClassOf) between A and C is 0.7*0.7 while A and B is 0.7).

The assignment of the weight factor is an interesting issue which needs more considerations. Currently we just give an arbitrary and somewhat "reasonable" static value manually. Static means the value doesn't change at runtime. It could be meaningful however if the value is adjustable during runtime. This can make the system more adaptive and realistic. The design of such an algorithm to adapt the factor values would be a challenging task. Besides of that, some heuristic approaches based on cognitive experiments can also be helpful to determine the factor value of each property.

Figure 17: The user model of MK within the context Context2

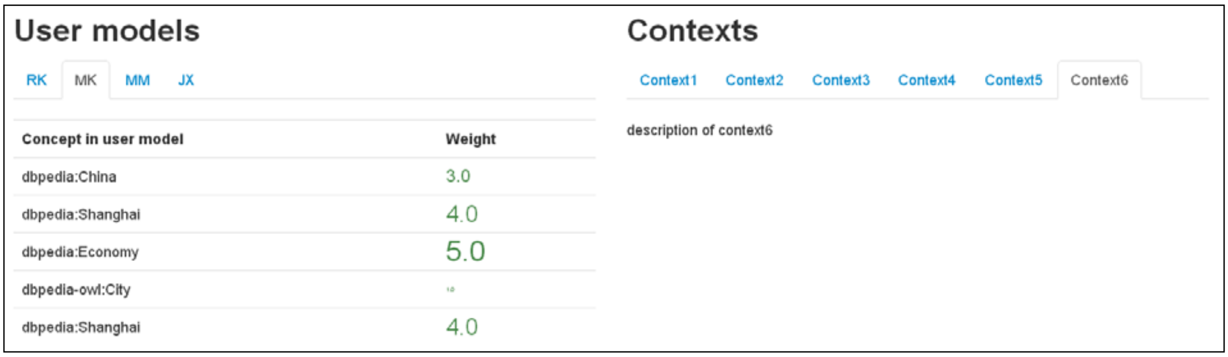


Figure 18: The user model of MK within the context Context6

10.3 Four Filtering Algorithms

Currently we have implemented four algorithms though some of them still need fine tuning. The inputs for all these four algorithms are the same: the Contextualized User Model CUM and the multimedia fragment annotation VA. The output of the filtering is a weighting and resulting ranking of multimedia fragment annotations.

The screenshot of a sample output is shown in Figure 20. For each multimedia fragment annotation element there are three attributes:

- v(URI) is the user-independent or video-specific weight of an annotation URI (denoted as W0 on the GUI). This value will be provided by WP2 (see D2.2). The weight v characterises the video content itself semantically.
- w(URI) is the user-specific weight of the URI (denoted as W1 on the GUI).
- TC is the time consumption for each calculation of w(URI). Wall time on the server is used to calculate w(URI). It is just an approximation and is only interesting for algorithm developers in order to fine tune the algorithms.

The four filtering algorithms implemented as part of LSF are described in more detail in the following subsections.

10.3.1 Precise matching

This is the most fundamental algorithm. It serves as the basis for the other three. In this algorithm, no “relatedTo” relation is considered. It just correlates the classes or instances in the CUM and the VA explicitly to each other.

It takes v(URI) for each multimedia fragment annotation element (URI), if the given user model contains explicitly this annotation, i.e. the user has interests in this annotation element (URI), then the product of both weights will be returned as W1 (see figures below). This approach has been explained in detail in Chapter 7.

Annotations of video3 0

Annotation	W0	W1	TC
dbpedia:Economy	4.0	n/a	n/a
dbpedia:Angela_Merkel	7.0	n/a	n/a
dbpedia-owl:City	9.0	n/a	n/a

Figure 19: A sample set of video annotations for video3

Annotations of video3 29

Annotation	W0	W1	TC
dbpedia:Economy	4.0	20.0	375.0 ms
dbpedia:Angela_Merkel	7.0	0.0	313.0 ms
dbpedia-owl:City	9.0	9.0	344.0 ms

Figure 20: User-specific weight of video3 is calculated for user MK

10.3.2 relatedTo-based filtering

In this algorithm all concrete “relatedTo” properties defined in Table 1 will be taken into account when the user model and the multimedia fragment annotations are matched.

This process can be very time consuming (since it involves querying the reference ontology which are normally available as RDF/SPARQL endpoint in the LOD cloud, see the architecture diagram in Figure 15) and involves almost always recursive operations. Therefore a threshold value, e.g. timeout or recursion levels, is normally needed to avoid endless recursion. This threshold values may depend on the concrete “relatedTo” property. For example, for the property “rdfs:subClassOf” a maximum depth-first search level can be given: if a superclass is too far away in the inheritance hierarchy, normally we can say they are less relevant and it is not necessary to performe time-consuming search operations.

After a “relatedTo” class or individual has been found, the precise matching procedure will be applied to get W1 or v(URI). All W1 are summarized[12] in the end to get an overall W1 for the given video annotation and user model.

10.3.3 WordNet-based

In this variant, WordNet [Fellbaum98] synonyms are used to find some related classes or individuals which have different URIs denotations but semantically “similar to” each other.

In current version, four types of synonyms in WordNet have been integrated into our system. They are adjective synonyms, adverb synonyms, noun synonyms and verb synonyms. After a synonym of a given video annotation has been found, it is then used to “construct” [13] another URI for further precise matching.

After all new URIs constructed from synonyms have been processed, the values are summarised as the overall W1 for the given video annotation.

10.3.4 Pattern-based

This algorithm is about using string patterns to find the “semantically” related classes or individual. We have investigated some URIs which are syntactically partially different but semantically quite similar to each other. In current system only substring patterns are applied. This is mainly based on the consideration of efficiency. It is a compromise between efficiency and effectiveness. This approach may loose some semantically similar entities, but comparing to the time it saves, it is good enough. A comprehensive experiment will be performed and analysed in the future.

10.3.5 Summary

All four algorithms are for the same task (calculate W1) but from different perspectives. It is difficult to say which one is better. It depends on different factors like user models, video annotations, etc. When to apply which one, or a combination of several algorithms is still unclear at the moment. We need some more real experiments and then evaluate the results based on human heuristic judgements. Therefore we provide an option in current system, so that the user can choose one or more to calculate W1 (see Figure 21).

[12] All W1 values are just summarised at the moment. More sophisticated and proper methods should be provided later depending on different creterias.

[13] In current system we just use the same prefix as the original one and append the found synonyms to create the new one. Later it can be meaningful to construct new URIs for different LoD ontologies.

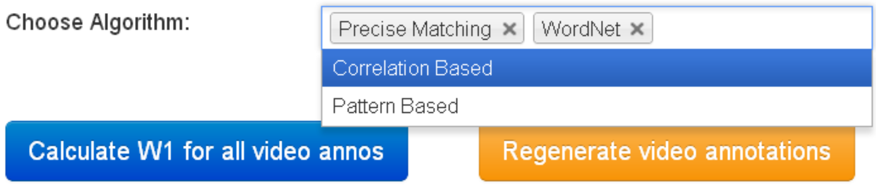


Figure 21: The option to choose one or more algorithms to get W1

10.4 Visualising the matching process

In order to debug and validate the system the matching process for a specific video annotation can be visualized as a tree (see Figure 22). This kind of visualisation is mainly for system developer instead of the end user. All the technical details will be hidden from end users in the future.

The root of the tree is always the video annotation which needs to be correlated or matched explicated with the user model. An option is provided to show the unmatched search path. Normally it is however not necessary to show the whole search process (it can be quite large and take considerable time to be loaded) but just the path on which a matching is found.

System developers can interactively in the single-page application change the visualisation based on the annotations being specified. All data necessary is loaded asynchronously via Ajax and JSON. Tree rendering operations are performed natively in the browser based on the D3.js visualisation framework. This ensures maximum system compatibility while conserving a high level of design flexibility and extensibility.

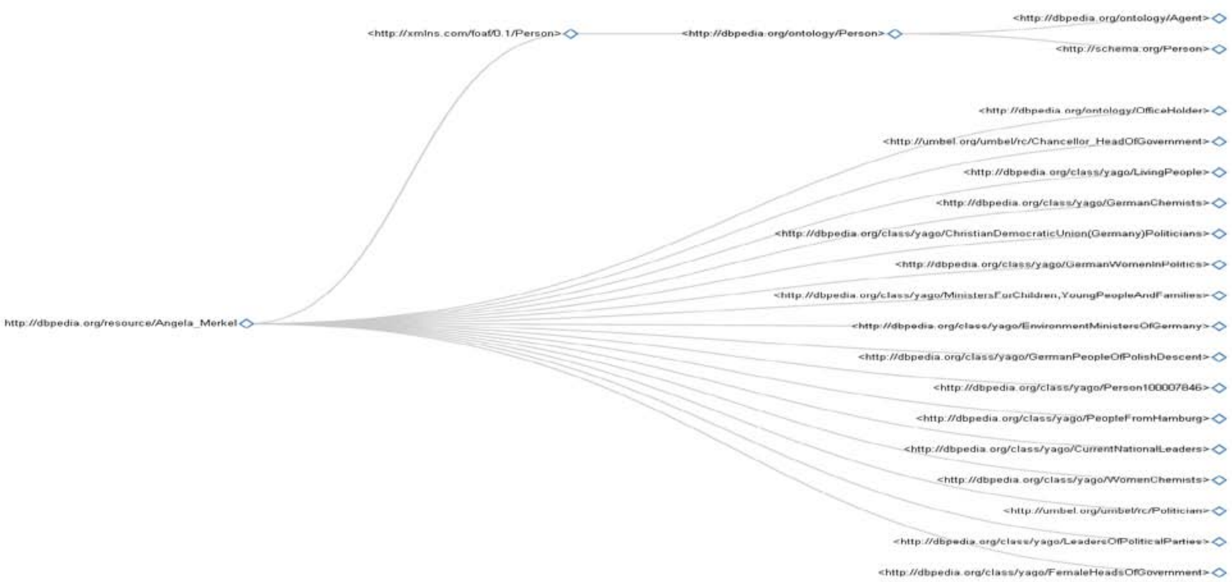


Figure 22: The search-and-correlation tree of annotation dbpedia:Angela_Merkel

11 Semantic reasoning post-filtering

A semantic reasoner is going to be employed to post-filter and re-rank the results of the first filtering algorithm LSF (described in the previous chapter). A semantic reasoner is an inference engine that derives logical consequences based on logical statements and asserted facts drawn from a semantic knowledge base. The benefit in using semantic reasoning for concept filtering and profile-content matchmaking lies in taking advantage of more complex knowledge than mere weighted concepts, like explicit axioms associating concepts, stemming from association rule mining (cf. D4.2). This axiomatic knowledge can reflect disinterest and association between preferences (e.g. interest in one concept only in conjunction with another, concept disjointness, negation etc).

Such an inferencing engine will also be able to make inference with formal reference knowledge – that being either future more well-structured or more domain-specific versions of the current LinkedTV ontology, or automatically learnt contextual complex knowledge for specific user clusters. The reason why it is proposed to be used for post-filtering lies on alleviating the input data load and complexity that hampers the processing time in most reasoning services, by minimizing the set of content items that the user profile needs to be matched against.

11.1 Implementation

The general idea of what a semantic reasoner is, thus what reasoning is (i.e. the globally applied algorithmic background for any given reasoner), is widely known in the semantic technologies world since reasoners have commonly been used as inferencing engines over semantic knowledge bases. This section will not delve into the details of the basics of reasoning because they constitute an entire scientific field (logic).

Following the basics, what needs to be described for any specific reasoner is:

1. the inferencing services it provides,
2. the semantics it supports that define the supported inference rules and
3. the "mechanics" employed to make use of those semantics to produce a conclusion.

The devised reasoner, f-PocketKRHyper, is an extension of the PocketKRHyper [KLE05]. PocketKRhyper, and consequently f-PocketKRHyper, is designed to be used in as limited in resource devices as previous generation mobile phones, thus laying the foundations for intelligent, automated knowledge processing *on limited resource devices as independently from external services as possible*. f-PocketKRhyper was initially extended within the aceMedia[14] EU project and since then has been further extended with an interest in addressing semantic profile matching.

PocketKRHyper is a (crisp) first-order logic (FOL) theorem prover that borrows semantics from Logic Programming (LP). f-PocketKRHyper follows the original implementation's theorem proving mechanics while extending it to fuzziness. Also, as in the original crisp reasoner, it implements LP-like semantics with the difference that f-PocketKRHyper is restricted to only the DLP [GRO03] fragment of FOL.

DLP is a tractable language and a subset of the expressivity covered in the original PocketKRHyper approach, while resulting in non-symmetrical use of DL constructors and axioms. The tradeoff between expressivity and complexity, thus performance, is the main reason that triggered the use of a tractable language.

PocketKRHyper additionally provided an interface for transforming Description Logic (DLs) to first order clausal logic (again with LP semantics, so for instance logical implication is not supported). f-PocketKRHyper has extended the original implementation's DL interface to compensate for missing DLP semantics (disjointness, negation added).

Additional semantics were implemented in order to provide support for uncertainty handling in both user preferences and concept assertions, based on Straccia's concept weight modifiers [BOB08] and Zadeh's fuzzy sets [ZAD96], respectively. Analytical tables of supported semantics can be found in the Appendix.

Consequently, by appending user model information and facts about either a given content item (for content filtering) or a set of related concepts (for concept filtering), formulated with formal semantics, in a given domain ontology, f-PocketKRHyper can decide whether the content item or concept(s) matches the user profile and to what degree.

The decision is based on whether or not the concepts in the concept set or the content item's semantic description satisfy the profile of the user, i.e. a synthetic concept "profile" (detailed description in D4.2) that subsumes all the concepts that the user is interested and disinterested in. The determination of the degree is based on the participation degree of fuzzy annotation metadata that accompany the content item[15] and modified by the weight indicating the user's level of interest/disinterest to the concept, following the [ZAD96] and [BOB08] principles.

[14] http://cordis.europa.eu/ist/kct/acemedia_synopsis.htm

[15] Participation in a set of related concepts are treated in the same manner, however currently in LinkedTV they are assumed to be crisp, thus coming with a degree of ≥ 1.

Main features

- Supports fuzziness in concept assertions. Crisp semantics supported as assertions with degree ≥ 1.
- Crisp role assertions.
- Weighted concepts
  - o Currently modifying a fuzzy concept assertion as the product of the concept weight and the degree of an individual participating in a concept [BOB08].
- Services implemented: *fuzzy entailment* and *subsumption,* used particularly for LinkedTV purposes for:
  - o Inferring the semantic model from the input knowledge base that satisfies the user profile (concept filtering), and
  - o Profile-to-content semantic matching (content filtering).
- Fuzzy entailment implemented as greatest lower bound (glb) calculation, i.e. when a model exists, the glbs of all fuzzy assertions are computed.
- Produced entailments are sound and complete only when within the DLP fragment. As a result the original reasoner's expressivity is restricted, and non-symmetrical axioms are imposed (i.e. $\forall R.A \subseteq B$ or $A \subseteq \exists R.B$ are not supported)).

11.2 Matchmaking

f-PocketKRHyper has proven to offer significant accuracy in semantic matchmaking between a given profile user profile to a set content items [TSA09]. Its accuracy nonetheless, like in any reasoning service, depends on the correctness and completeness of the input data, i.e. the reference knowledge base, the annotation and the user profile.

Toy example

The following fabricated example demonstrates at an abstract level the matchmaking process for 3 cases of annotated content given user interests in a news sub-domain. The semantics of the reference knowledge are also fabricated for demonstration purposes and do not necessarily reflect exactly the actual LinkedTV user model ontology LUMO. However in the context of LinkedTV the reference ontology used is going to be the dedicated WP4 User Model Ontology and the axiomatic user model created through implicit preference learning (cf. D4.2).

User Profile:

∃hasInterest.(0.95·Rule1 ∪ 0.77·PoliticalEvent(Election)) ⊆ Interests

(either satisfies the profile)

0.87 ·Politics ∩ 0.59·Politician(Bill Clinton) ⊆ Rule1

(interested in politic topics only in relation to Bill Clinton)

∃hasInterest.(0.72·Politician(Barack Obama)) ⊆ Disinterests

(does not in any case want to view content about Barack Obama)

Reference knowledge:

Politician ⊆ Politics

PoliticalEvent ⊆ Politics

Profile explicit constructors and ABox:

Interests ∩ Disinterests ⊆ ⊥

(disjoint concepts)

<inst,Y>: hasInterest

(fabricated role assertion)

(inst: indivisual instantiating the profile; Y: variable instantiated by all of the individuals annotating concepts in the content annotation).

Annotation in related content items to recommend:

A) <Al Gore>: Politician ≥ 0.8
B) <Bill Clinton>: Politician ≥ 0.65
C) <Election>: PoliticalEvent ≥ 0.76, <Barack Obama>: Politician ≥ 0.89

Result:

	Inferred Model	Satisfies interest/disinterest	Match
A	Politician(Al Gore) ≥ 0.8 Politics(Al Gore) ≥ 0.696 hasInterest(inst, Al Gore)	-	N
B	Politician(Bill Clinton) ≥ 0.65 Politics(Bill Clinton) ≥ 0.5655 hasInterest(inst, Bill Clinton) Rule1(inst) ≥ 0.5655	Interests(inst) ≥ 0.5655	Y
C	PoliticalEvent(Election) ≥ 0.5852 Politics(Election) ≥ 0.66	Interests(inst) ≥ 0.5852 Disinterests(inst) ≥ 0.64	N

	`Politician(Barack Obama) ≥ 0.64`
	`Politics(Barack Obama) ≥ 0.77`
	`hasInterest(inst, Barack Obama)`
	`hasInterest(inst, Election)`
		→
		Refutation	

11.3 Cooperation of the filtering algorithms and reasoner benefits

In a nutshell, the reasoner will be post-filtering the LSF results with an aim to improve them. The improvement is a direct consequence from the fact that the reasoner can handle additional, more complex knowledge that the LSF cannot take into account. Post-filtering is opted because a) the LSF is expected to be fast, i.e. quickly produce initial comprehensive results, and to minimize the input data space, while b) the reasoner is slower since it takes into account more complex knowledge, which raises the complexity of the inference problem (implying slower delivery of results) but will provide slim, precise and meaningful results that can take into account the added knowledge and also detect inconsistencies in the initial results.

In addition, the capability of f-pocketKRhyper to run in limited resource devices allows for the inferencing and recommendation process to take place *either on the server or on the end-device* seamlessly and effectively, thus enabling additional privacy preservation possibilities in concept and content filtering. Nonetheless, applying the reasoning process in an end-(limited resource)-device depends directly on the size and complexity of the input knowledge base. Even as the design of the user model and the reference ontology are for this reason kept as lightweight as possible (cf. D4.2), the concept space in the super-domain of networked media is still considerably large. In the future however, it is expected that the background knowledge can be significantly reduced (contextual profiles, contextual reference knowledge subsets pulling) as will the problem space (related concepts and content) be reduced based on the results of the first filtering processes.

11.4 Current and future extensions

Extensions in the scope of LinkedTV are considered in order to address required fuzzy semantics stemming from the developed preference learning algorithms. Such extensions include the introduction of weighted sum and threshold semantics [BOB08] as concept modifiers and conditional axioms support. In the future, implementation of fuzzy role assertions will also be considered, as will the concept assertion degree calculating functions be reconsidered in order to comply with LinkedTV requirements.

12 Summary and Outlook

In this document we outlined a first approach to content filtering based on semantic user models and matching these models and semantic multimedia annotations.

This approach is based on a couple of assumptions and restrictions. The most important ones are the following:

- The user model we are using to represent a user's special interests and preferences is based on an ontology. We used our own LinkedTV user model ontology LUMO as starting point for semantic user models (as described in D4.2). It comprises the main kinds of knowledge a user has about the world and his interests in it. It includes the main concepts and topics as well as concrete instances a user has special interest in. The LUMO provides the bridge to other existing LOD ontologies to be used in multimedia annotations (see D2.2). The user model ontology used in this document is just "work in progress". It may change in various directions in the future. Any other ontology which is able to represent the user model can be used in a similar way in our filtering approach.
- There are different ways to get and to maintain a user model (see D4.2 for details). The user model is designed to contain those items the user has a special interest in including its semantic generalizations (super concepts). It does *not* contain anything the user did not show a special interest in.
- Each term in the user model ontology (concepts as well as concrete instances) can be assigned a weight expressing the user's interest in this term. How these weights are gained is described in more detail in D4.2. The filtering approach described in this document is completely flexible w.r.t. the weights used in the user models.
- The user media consumption will provide a lot of information about the user's preferences. Some of these insights can be represented in the user model through adapted weights. In other cases, a modification of the user model ontology can be useful – for instance the introduction of new concepts which better represent the user's interests. If these concepts exist already in reference ontologies and are used there to classify entities their integration into the user model is straightforward. If this is not the case more sophisticated concept learning and reasoning techniques will be needed.
- The inferences used to relate user models and multimedia annotations can be very complex. In the first filtering version LSF V0.1 described in this document we deliberately focused onto an efficient algorithm. It is based on semantic mappings using type and subclassOf relations. Experiments will be needed to find out which extensions are needed and feasible.
- The mapping from video annotations to user models is straightforward as long as the reference ontology of the annotation element is connected to the user model ontology. These mappings are a simple form of inferences based on type and subclassOf relations. They provide an efficient way to match user models with

multimedia annotations. If the reference ontology of the video annotation is not linked to the LUMO we apply ontology alignment techniques with *heuristic* mappings.

- In this document, we discussed a couple of alternatives how to treat various issues: weight factors for subclassOf and relatedTo relations, how a consumed video can change the user model weights, etc. The whole user modeling and filtering approach described here is experimental by nature. It will need validation by tests and experiments.
- The f-PocketKRHyper reasoner allows us to post-process LSF filter results using knowledge represented as fuzzy DLP axioms as parts of user models.

The content filtering approach described in this document is a first version. Using it in the project will help us to extend it along various directions in the future:

- Semantic content filtering is based on three aspects: the user models, the multimedia objects (with their annotations and semantic extensions), and the user interactions. The better they are adapted to each other the better they can work together. Within the LinkedTV project results from different work packages have to be brought together for this purpose.
- Improved user model ontologies: new developments in the Linked Open Data world may result in better ontologies with better relations to each other; size and structure of user models following concrete experiences in the project; how to build user models and how to maintain them, etc.
- The filtering approach introduced here is based on light weight ontologies and weights for concepts and instances. It applies a simple weighted semantic matching technique. This is an effective and efficient approach usable in many cases. More sophisticated filtering techniques are needed if higher precision of user modeling and semantic filtering is needed. This includes various reasoning techniques like f-PocketKRHyper. An integrated implementation of LSF and f-PocketKRHyper will be used to combine the advantages of both techniques in practice.
- Adequate annotations of multimedia content are a pre-requisite for efficient user support. The better these annotations can be aligned with the user models the better the guidance we can provide to the user. This means to identify the concrete entities in a multimedia object with their relations to each other (including temporal and spatial aspects), and to assign them to their concepts according to the concepts used in the user model. This also means to provide reliable and efficient semantic mappings from these annotations to the user model.
- Our current filtering procedure applies a simple context approach: it simply selects parts of the user model as current context. In the future we will work on more sophisticated contextualisations which take typical user behavior into account.
- User interactions are a key issue in media consumption. The semantic content filtering introduced here can be applied in different ways to support the user. In a concrete video frame the user may get a ranked presentation of media items just shown; he may get additional information ranked according to his preferences to a

media object he just selected; or he may get recommendations for similar media content. The semantic filtering supports all these usages. The integration of the content filtering with the user interaction capabilities will allow us to work and to experiment with these features.

- Observing the user behavior (his media selection behavior as well as his comments, facial reactions, communications etc.) can be used to draw conclusions about the quality of the recommendations generated by the system. They will allow us to adapt the user model accordingly and to improve it for future usages.

13 References

[Alia06] Mourad Alia, Rolf Fricke, Geir Horn, Frank Eliassen, Mohammad Ullah Khan, Roland Reichle: A Component-Based Planning Framework for Adaptive Systems. OTM Conferences (2) 2006: 1686-1704.

[Andrej06] Andrejko, A., Barla, M., und Bielikov, M. (2006). Ontology-based user modeling for web-based information systems. In: Maygar, G., Knapp, G., Wojtkowski, W., Wojtkowski, G., und Zupancic, J. (Hrsg.), Advances in Information Systems Development: New Methods and Practice for the Networked Society, Band 2, Berlin. Springer.

[Anton08] Antoniou, G. und van Harmelen, F. (2008). A Semantic Web Primer, second edition. The MIT Press.

[BOB08] Fernando Bobillo and Umberto Straccia. fuzzyDL: An Expressive Fuzzy Description Logic Reasoner. In Proceedings of the 2008 International Conference on Fuzzy Systems (FUZZ-08), 2008.

[Brajnik, G. and Tasso, C] Brajnik, G. and Tasso, C.: A Shell for Developing Non-monotonic User Modeling Systems. International Journal of Human-Computer Studies 40, (1994) 31-62, DOI 10.1006/ijhc.1994.1003.

[Brusilovsky, P] Brusilovsky, P.: KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), New York, NY (2004) 104-113, 10.1145/1013367.1013386.

[Brusilovsky, P. and Millán, E, Henze, N.] Brusilovsky, P. and Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: The Adaptive Web: Methods and Strategies of Web Personalization, Brusilovsky, P., Kobsa, A., and Nejdl, W., Eds. Berlin Heidelberg New York: Springer Verlag (2007) this volume.

[Brusilovsky, P., Sosnovsky, S., and Yudelson, M] Brusilovsky, P., Sosnovsky, S., and Yudelson, M.: Ontology-based Framework for User Model Interoperability in Distributed Learning Environments. World Conference on ELearning, E-Learn 2005, Vancouver, Canada (2005) 2851-2855, http://www2.sis.pitt.edu/~peterb/papers/eLearn2005-adapt.pdf.

[Carmichael, D. J., Kay, J., and Kummerfeld, B] Carmichael, D. J., Kay, J., and Kummerfeld, B.: Consistent Modelling of Users, Devices and Sensors in a Ubiquitous Computing Environment. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 15, (2005) 197-234, DOI 10.1007/s11257-005-0001-z.

[Fellbaum98] C. Fellbaum. WordNet : An Electronic Lexical Database. The MIT Press, Cambridge, MA, 1998.

[Finin and Drager 1986] Tim Finin and David Drager. GUMS: a General User Modelling System. In Proceedings of the Canadian Society for Computational Studies of Intelligence 1986 (CSCSI-86), 1986. http://acl.ldc.upenn.edu/H/H86/H86-1021.pdf [Last access June 13th, 2005].

[Fink, 2004] Fink, J. (2004). *User Modeling Servers - Requirements, Design, and Evaluation*, volume 289 of *DISKI*. Aka Verlag, Berlin. ISBN 3-89838-277-X.

[Fink and Kobsa, 2000] Fink, J. and Kobsa, A. (2000). A review and analysis of commercial user modeling servers for personalization on the world wide web. *User Modeling and User-Adapted Interaction*, 10(2-3):209–249.

[Fink and Kobsa, 2002] Fink, J. and Kobsa, A. (2002). User modeling for personalized city tours. *Artificial Intelligence Review*, 18(1):33–74.

[Geihs09] Kurt Geihs, Rolf Fricke, et al: A comprehensive solution for application-level adaptation. Software, Practical Experiments, 39(4): 385-422 (2009).

[GRO03] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. 2003. Description logic programs: combining logic programs with description logic. In Proceedings of the 12th international conference on World Wide Web (WWW '03). ACM, New York, NY, USA, 48-57.

[Heck05] Heckmann, D. (2005). Ubiquitous User Modeling. PhD thesis, Universität des Saarlandes, Saarbrücken.

[Henze07] Henze, N. and Brusilovsky, P.: Open Corpus Adaptive Educational Hypermedia. In: The Adaptive Web: Methods and Strategies of Web Personalization, Brusilovsky, P., Kobsa, A., and Nejdl, W., Eds. Berlin Heidelberg New York: Springer Verlag (2007).

[Jann12] Dietmar Jannach, Markus Zanker, Alexander Felfering, Gerhard Friedrich: Recommender System – An introduction (2012); Cambridge University Press

[Kay, J 90] Kay, J.: UM: A Toolkit for User Modelling. In: Second International Workshop on User Modeling. Honolulu, HI (1990) 1-11.

[Kay, J 95] Kay, J.: The um Toolkit for Reusable, Long Term User Models. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 4, (1995) 149-196, DOI 10.1007/BF01100243.

[Kay, J 99] Kay, J.: A Scrutable User Modelling Shell for User-Adapted Interaction. In Basser Department of Computer Science: University of Sydney, Australia (1999), http://www.cs.usyd.edu.au/~judy/Homec/Pubs/thesis.bz2.

[Kay, 1995] Kay, J. (1995). The UM toolkit for reusable, long term user models. *User Modeling and User-Adapted Interaction*, 4(3):149–196.

[Kay, J., Kummerfeld, B., and Lauder, P] Kay, J., Kummerfeld, B., and Lauder, P. (2002). Personis: A server for user models. In de Bra, P., Brusilovsky, P., and Conejo, R., editors, *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 2347, pages 203–212. Springer-Verlag Berlin Heidelberg.

[Kay02] Kay, J., Kummerfeld, B., and Lauder, P.: Personis: A Server for User Models. In: Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002, De Bra, P., Brusilovsky, P., and Conejo, R., Eds. Berlin Heidelberg: Springer (2002) 203–212, http://springerlink.metapress.com/link.asp?id=2l54yrgc0p8n2d5g.

[KLE05] Kleemann T., Sinner A. 2005. User Profiles and Matchmaking on Mobile Phones. In proceedings, INAP 2005, 16th International Conference on Applications of Declarative Programming and Knowledge Management, Fukuoka 2005.

[Kobsa 2001a] Alfred Kobsa. Generic User Modeling Systems. User Modeling and User-Adapted Interaction, vol. 11, no. 1-2 p.p. 49–63, 2001. http://www.ics.uci.edu/%7Ekobsa/papers/2001-UMUAI-kobsa.pdf [Last access May 25th,2005].

[Kobsa, A. and Pohl, W.] Kobsa, A. and Pohl, W.: The BGP-MS User Modeling System. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 4, (1995) 59-106, DOI 10.1007/BF01099428.

[Kray, 2003] Kray, C. (2003). *Situated Interaction on Spatial Topics*, volume 274 of *DISKI*. Aka Verlag, Berlin. ISBN 1-58603-391-3.

[Malaka and Zipf, 2000] Malaka, R. and Zipf, A. (2000). DEEP MAP - challenging it research in the framework of a tourist information system. In Fresenmaier, D., Klein, D., and Buhalis, D., editors, *Information and Communication Technologies in Tourism 2000*, pages 15–27. Springer.

[Mobasher, B] Mobasher, B.: Data Mining for Web Personalization. In: The Adaptive Web: Methods and Strategies of Web Personalization, Brusilovsky, P., Kobsa, A., and Nejdl, W., Eds. Berlin Heidelberg New York: Springer Verlag (2007).

[Murray87] Dianne M. Murray. Embedded User Models. In Proceedings of the 2nd IFIP International Conference on Human-Computer Interaction (INTERACT' 87), P.p. 229–235, 1987.

[NoTube] NoTube: D3.3 Recommender Service, Deliverable of the EU Project NoTube, www.notube.tv

[Orwant, J] Orwant, J.: Heterogenous Learning in the Doppelänger User Modeling System. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 4, (1995) 107-130, DOI 10.1007/BF01099429.

[Orwant, 1994] Orwant, J. (1994). Apprising the user of user models: Doppelg¨anger's interface. In *Proceeding of the Forth International Conference on User Modeling*, pages 151–156, Massechusetts, USA. MITRE, User Modeling Inc, Hyannis.

[Orwant, 1995] Orwant, J. (1995). Heterogeneous learning in the Doppelg¨anger user modeling system. *User Modeling and User-Adapted Interaction*, 4(2):59–106.

[Orwant, 1996] Orwant, J. (1996). For want of a bit the user was lost: Cheap user modeling. *IBM Systems Journal*, 35(348):398–416.

[Paiva, A. and Self, J.] Paiva, A. and Self, J.: TAGUS: A User and Learner Modeling System. In: Proc. of the Fourth International Conference on User Modeling. Hyannis, MA (1994) 43-49.

[Pohl 1999] Pohl, W.: Logic-Based Representation and Reasoning for User Modeling Shell Systems. User Modeling and User-Adapted Interaction: The Journal of Personalization Research 9, (1999) 217-282, DOI 10.1023/A:1008325713804.

[Rich79] Elaine Rich. User Modeling via Stereotypes. Cognitive Science, vol. 3, p.p. 329–354, 1979. http://www.cs.utexas.edu/users/ear/CogSci.pdf [Last access June 21th, 2005].

[Ricci11] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor: Recommender Systems Handbook; Springer Verlag; 2011

[Sieg07] Sieg, A., Mobasher, B., und Burke, R. (2007). Web search personalization with ontological user profiles. In: Silva, M. J., Laender, A. H. F., Baeza-Yates, R. A., McGuinness, D. L., Olstad, B., Olsen, Ø. H., und Falc˜ao, A. O. (Ed.), Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007, S. 525–534, New York, NY, USA. ACM.

[TSA09] D. Tsatsou, F. Menemenis, I. Kompatsiaris and P. C. Davis. A Semantic Framework for Personalized Ad Recommendation based on Advanced Textual Analysis. *3rd ACM Conference on Recommender Systems (RecSys'09)*, NY, USA, 2009.

[Vergara, H.] Vergara, H.: PROTUM: A Prolog Based Tool for User Modeling. WG Knowledge-Based Information Systems, Department of Information Science, University of Konstanz, Germany, WIS-Report 10, (1994).

[Yoo12] Kyung-Hyan Yoo, Ulrike Gretzel, Markus Zanker: Persuasive Recommender Systems – Conceptual Background and Implications; Springer Verlag; ISBN 978-1-4614-4701-6

[ZAD96] Zadeh, L. A.. Fuzzy Sets, Fuzzy Logic, Fuzzy Systems. Advances in Fuzzy Systems — Applications and Theory: Volume 6, World Scientific Press, 1996. ISBN 9810224214

14 Appendices

14.1 Appendix-1: Example User Characters

The following is an outline of six characters defined in WP6 as examples for prototypical users and used in Chapter 8 for demonstration purposes:

Lukas / German Single Male

- Age: 35
- Ethnicity: Caucasian
- Marital Status: Single (lives alone)
- Languages: German, English, French
- Sex: Male
- Occupation: Material Engineer. Lukas works for Thyssen Krupp AG.
- Technical Background: Definitely technical. Likes material science and physics. Has had mechanical engineering and material science at university
- Educational status: Masters degree in mechanical engineering and material science
- Income Bracket: 45,000 – 55,000 €
- Technology owned and used: PC every day for work (MS office applications, web-browsing, e-Mail, engineering applications). Does basic system administration of the home laptop and uses it for email, IM, Facebook. He has a 3 years old Samsung mobile phone and uses it for making phone calls and SMS, nothing else. He has an NIKON digital camera which he uses to collect footage when on vacation. He owns one old TV (not HD ready) and DVD player. He has no need for recording devices.
- Socio-economic status: Middle Class Professional
- Family: Has one younger sister
- Hobbies: He has not much time for hobbies but enjoys spending time with family or friends, sometimes watches TV and films, visits music acts, football.
- Favorite TV programs: German “Bundesliga” (different soccer clubs), Boxing events, Scrubs (american hospital serie), documentaries about astrophysics (black holes, super novae, extra-terrestric species)
- Favorite Movies: Rambo (complete film series), The Matrix, Avatar
- Favorite Music: Hard Rock from the 90s till today
- Other audio programs: Local commercial radio for news, traffic and sport, listen to while driving to work
- Other interests: He was one year in military duty. So, he is interested in the entire military technology
- Concerns: He is very money-conscious because he is concerned about his job safety. But he saves money to replace his 10 years old car soon.

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Material Engineer. Lukas works for Thyssen Krupp AG	**ThyssenKruppAG** as instanceOf Organisation		http://dbpedia.org/resource/ThyssenKrupp	10
Definitely technical. Likes material science and physics. Has had mechanical engineering and material science at university	**Technology** as subClassOf Topic, **MaterialScience** as instanceOf Engineering, **Physics** as subClassOf Science, **Mechanical Engineering** as instanceOf Engineering, **University** as instanceOf EducationalOrganisation	**Physics** as subClassOf Science, **Technology** as subClassOf Topic, **Engineering** as subClassOf Technology	http://linkedtv.eu/topic/technology http://linkedtv.eu/science/physics http://linkedtv.eu/topic/technology/engineering http://dbpedia.org/resource/university http://dbpedia.org/resource/materialscience http://dbpedia.org/resource/mechanicalengineering	10 10 10 5 5 5
He has a 3 years old Samsung mobile phone and uses it for making phone calls and SMS, nothing else. He has an NIKON digital camera which he uses to collect footage when on vacation.	**Samsung** as instanceOf Organisations **NIKON** as instancOf Organisations		http://dbpedia.org/resource//Samsung http://dbpedia.org/resource//NIKON	5 5
He has not much time for hobbies but enjoys spending time with family or friends, sometimes watches TV and films, visits music acts, football.	**TV** as subClassOf Topic, **Films** as similarTo Movies, **MusicAct** as similarTo **MusicEvent**, **Football** as subClassOf Sports	**TV** as subClassOf Topic, **Films** as similarTo Movies, **MusicAct** as similarTo **MusicEvent**, **Football** as subClassOf Sports	http://linkedtv.eu/topic/tv http://linkedtv/topic/movie http://schema.org/musicEvent http://linkedtv/topic/sports/football	10 10 10 10
German "Bundesliga" (different soccer clubs), Boxing events, Scrubs (american hospital series), documentaries about astrophysics (black holes, super novae, extraterrestric species)	**Bundesliga** as instanceOf SportsLeague, **BoxingEvents** as instance of SportsEvent, **Documentaries** as subClassOf Documentation, **Astrophysics** as subclassOf Physics, **BlackHoles** as instanceOf Astrophysics,	**SportEvents** as subClassOf Events, **documentaries** as subclassOf TV, **astrophysics** as subClassOf Physics	http://dbpedia.org/resource/Bundesliga http://dbpedia/resource/Boxing http://linkedtv.eu/topic/tv/documentation http://linkedtv.eu/topic/science/physics/astrophysics http://dbpedia.org/resource/blackhole http://dbpedia.org/resource/supernova http://dbpedia.org/resource/extraterrestricspecies	5 5 10 10 5 5 5

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
	SuperNovae as instanceOf Astrophysics, **Extraterrestric Species** as instanceOf Astrophysics		http://schema.org/event/sportsevent	10
Rambo, The Matrix, Avatar	**Rambo** as instanceOf Movie, **TheMatrix** as instanceOf Movie, **Avatar** as instanceOf Movie	**Movie** as subClassOf Topic	http://dbpedia.org/resource/Rambo http://dbpedia.org/resource/TheMatrix http://dbpedia.org/resource/Avatar http://linkedtv.eu/topic/movie	5 5 5 10
Hard Rock from the 90s till today	**HardRock** as subClassOf MusicGenre	**Music** as subClassOf topic, **HardRock** as subClassOf MusicGenre	http://dbpedia/resource/HardRock http://linkedtv.eu/topic/music	10 10
Local commercial radio for news, traffic and sport, listen to while driving to work	**News** as subClassOf TV, **sport** as subClassOf Report	**News** as subClassOf TV, **sport** as subClassOf Report	http://linkedtv.eu/topic/tv/news http://linkedtv.eu/topic/tv/report	10 10

Sophie / Dutch Student

- Age: 20
- Ethnicity: African descent
- Marital Status: Single
- Languages: Dutch, English, Spanish
- Sex: Female
- Occupation: Student of the lectureship of history and arts and waitress in a student-bar every evening
- Technical Background: No formal technical training, basic computer skills
- Educational status: Bachelor degree
- Income Bracket: Student-job and parents support. Monthly income 800€.
- Technology owned and used: She uses private MacBook Air every day for study (MS office applications, web-browsing, e-Mail) and photo-processing. She owns a high-end digital camera and uses it for nature and architecture photography, sometimes her pictures are published in local news magazines. She has a 32” flat-screen TV (HD ready). She listens to MP3 music on iPod during sports-activities. Has a modern mobile phone every two years and uses it for calling, text messaging, blogging and sometimes geo-caching during holiday trips
- Socio-economic status:
- Family:
- Hobbies: photography, fitness-training (gym), travelling abroad, going out (nightlife)
- Favorite TV programs: usually watches news, otherwise watches documentaries on nature and history (History channel), soap operas, How I met your mother (American series)
- Favorite Movies: Dirty Dancing, Twilight, Ice Age
- Favorite Music: Rock and Pop from the 90s up to present day, comprehensive collection derived from iTunes and Spotify.
- Other audio programs: Local commercial radio chart music on the way to and from university in car
- Other interests: Environment, politics, society, lifestyle, socializing with wide network of friends from school and university via Facebook
- Concerns: Although happy about online-shopping, concerned about data privacy and is careful to protect her identity where possible. She lives eco-sensitive and turns off un-used appliances and avoids garbage.

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Student in the lectureship of history and arts		**History** as subClassOf Topic, **Arts** as subClassOf Topic	http://linkedtv.eu/topic/history http://linkedtv.eu/topic/arts	10 10
She uses private MacBook Air every day for study (MS office applications, web-browsing, e-Mail) and photo-processing. She owns a high-end digital camera and uses it for nature and architecture photography, sometimes her pictures are published in local news magazines. She has a 32" flat-screen TV (HD ready). She listens to MP3 music on iPod during sports-activities. Has a modern mobile phone every two years and uses it for calling, text messaging, blogging and sometimes geo-caching during holiday trips	**MacBookAir** as instanceOf Product, **iPod** as instanceOf Product	**Architecture** as subClassOf Arts, **Nature** as subClassOf Topic, **Photography** as SubClassOf Hobbies, **Holiday** as subClassOf Hobbies, **Geocaching** as subClassOf Hobbies	http://dbpedia.org/resource/macbookair http://linkedtv.eu/product http://linkedtv.eu/topic/arts/architecture http://linkedtv.eu/topic/nature http://linkedtv.eu/topic/hobbies/photography http://dbpedia.org/resource/ipod http://linkedtv.eu/topic/hobbies/holiday http://linkedtv.eu/Topic/Hobbies/Geocaching	5 5 10 10 10 5 10 10
photography, fitness-training (gym), travelling abroad, going out (nightlife)		**Photography** as SubClassOf Hobbies, **Gymnastics** as subclassOf Sports, **Travel** as SubClassOf Hobbies, **Nightlife** as SubClassOf Hobbies	http://linkedtv.eu/topic/hobbies/photography http://linkedtv.eu/topic/sports/gymnastics http://linkedtv.eu/topic/hobbies/travel http://linkedtv.eu/topic/hobbies/nightlife	10 10 10 10
Favorite TV programs: usually watches news, otherwise watches documentaries on nature and history (History channel), soap operas, How I met your mother (American series)	**HowIMetYourMother** as instanceOf Series, **HistoryChannel** as instanceOf Documentation	**News** as subClassOf TV, **Documentation** as subClassOf TV, **SoapOpera** as subClassOf Series,	http://linkedtv.eu/topic/tv/news http://linkedtv.eu/topic/tv/documentation http://linkedtv.eu/topic/tv/serie/soapopera http://dbpedia.org/resource/HowIMetYourMother http://dbpedia.org/resource/HistoryChannel	10 10 10 5 10

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Favorite Movies: Dirty Dancing, Twilight, IceAge	**DirtyDancing** as instanceOf Movie, **Twilight** as instanceOf Movie, **IceAge** as instanceOf Movie	**Movie** as subClassOf Topic	http://dbpedia.org/resource/DirtyDancing http://dbpedia.org/resource/Twilight http://dbpedia.org/resource/IceAge http://linkedtv.eu/topic/movie	5 5 5 10
Rock and Pop from the 90s up to present day, comprehensive collection derived from iTunes and Spotify.	**iTunes** as instanceOf ??, **Spotify** as instanceOf ??	**Music** as subClassOf topic, **Rock** as subClassOf MusicGenre, **Pop** as subClassOf MusicGenre	http://dbpedia.org/resource/Rock http://dbpedia.org/resource/Pop http://linkedtv.eu/topic/music http://dbpedia.org/resource/iTunes http://dbpedia.org/resource/Spotify	10 10 10 5 5
Other interests: Environment, politics, society, lifestyle, socializing with wide network of friends from school and university via Facebook	**Facebook** as instanceOf NewMedia	**Environment** as SubClassOf Topic, **Politics** as SubClassOf Society,	http://linkedtv.eu/topic/environment http://linkedtv.eu/topic/society/politics http://dbpedia.org/resource/facebook	10 10 5
Although happy about online-shopping, concerned about data privacy and is careful to protect her identity where possible. She lives eco-sensitive and turns off un-used appliances and avoids garbage.				

Jose / Spanish elder

- Age: 62
- Ethnicity: Caucasian
- Marital Status: Married
- Languages: Spanish, little English
- Sex: Male
- Occupation: medium enterprise owner
- Technical Background: Uses technology but is not very interested in it.
- Educational status: Diploma
- Income Bracket: 80,000-100,000€/year
- Technology owned and used: Home cinema, Blue Ray and High-End sound system, mobile phone, PC(Internet)
- Socio-economic status: Mid-upper class
- Family: Wife, two daughters and one son
- Hobbies: reading, classic music, going out (restaurant)
- Favorite TV programs: Political, social and financial debates, CSI series
- Favorite Movies: Western, action films
- Favorite Music: Beethoven, Mozart, Bach
- Other audio programs: Local commercial radio chart music on way to and from university in car
- Other interests: He likes to travel (camping and all-inclusive hotel), works in his garden and collects stamps.
- Concerns: Although happy about online-shopping, concerned about data privacy and is careful to protect his identity where possible. He lives eco-sensitive and turns off un-used appliances and avoids garbage.

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Technology owned and used: Home cinema, Blue Ray and High-End sound system, mobile phone, PC(Internet)	**HomeCinema** as instanceOf Technology, **BlueRay** as instanceOf VideoObject		http://dbpedia.org/ressoruce/homecinema http://schema.org/mediaobject/videoobject/blueray	5 5
Hobbies: reading, classic music, going out (restaurant)		**Reading** as subClassOf Hobbies, **ClassicMusic** as subClassOf MusicGenre, **Restaurant** as subClassOf Hobbies	http://linkedtv.eu/topic/hobbies/Reading http://dbpedia.org/resource/classicmusic http://linkedtv.eu/topic/hobbies/restaurant	10 10 10

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Favorite TV programs: Political, social and financial debates, CSI series	**CSI** as instanceOf TVSeries	**Politics** as subClassOf Society, **Society** as SubClassOf Topic, **Finance** as instanceOf Economy	http://linkedtv.eu/topic/society/politics http://linkedtv.eu/topic/society http://dbpedia.org/resource/finance http://dbpedia.org/resource/csi	10 10 10 5
Favorite Movies: Western, action films		**Movie** as subClassOf Topic, **Western** as subClassOf Movie, **Action** as subClassOf Movie	http://linkedtv.eu/topic/movie http://movieontology/genre/Western http://movieontology/genre/Action	10 10 10
Favorite Music: Beethoven, Mozart, Bach	**Beethoven** as instanceOf Classic, **Mozart** as instanceOf Classic, **Bach** as instanceOf Classic	**Music** as subClassOf Topic,	http://dbpedia.org/resource/Beethoven http://dbpedia.org/resource/Mozart http://dbpedia.org/resource/cBach http://linkedtv.eu/topic/music	5 5 5 10
Other audio programs: Local commercial radio chart music on way to and from university in car		**ChartMusic** as similarTo PopMusic	http://dbpedia.org/resource/Pop	10
Other interests: He likes to travel (camping and all-inclusive hotel), works in his garden and collects stamps.	**Stamps** as instanceOf Collections	**Travel** as subClassOf Hobbies, **Camping** as subClassOf Hobbies, **Gardening** as subClassOf Hobbies, **Collection** as subClassOf Hobbies	http://linkedtv.eu/Topic/Hobbies/Travel http://linkedtv.eu/Topic/Hobbies/Camping http://linkedtv.eu/Topic/Hobbies/Gardening http://dbpedia.org/resource/Stamp http://linkedtv.eu/Topic/Hobbies/Collection	10 10 10 5 10
Concerns: Although happy about online-shopping, concerned about data privacy and is careful to protect her identity where possible. She lives eco-sensitive and turns off un-used appliances and avoids garbage.				

Ralph / Sports loving carpenter

- Name and occupation: Ralph, carpenter
- Age: 19
- Nationality / place of residence: German / Prenzlau (Brandenburg)
- Digital literacy: digital native
- Main interests: Prenzlau and surroundings, architecture, nature, joinery, crafts, local sport clubs, boats, cars, car mechanics

Ralph has always lived in Prenzlau. After school he served an apprenticeship as carpenter in the neighbor village Strehlow. When he turned 18, he immediately bought his own car. Since then he has been tuning and improving it every weekend. A few weeks later, he met Cindy, a high school girl. They fell in love quickly and became a couple. Soon Ralph moved into his own apartment near the lake. Ralph and Cindy, who recently turned 18, have just celebrated their first anniversary. Both spend much time together and like to go for a walk. In summer they often go swimming in the nearby lake. Then Ralph is looking at the boats too, often thinking it would be nice to have one too one day, but he will have to learn to sail first. They love nature and look forward to moving into their own house near the forest one day. For Ralph it is a great idea to work in his own house. Being a carpenter, he is very interested in materials, building fabric, constructions and architecture. On the other hand, he is very interested in all local sports clubs. He does not care particularly for the latest technology, but as a “digital native” it is not difficult for him to adapt to new technologies and to use them.

Tuesday afternoon, Ralph comes home from working on a building site in Potsdam. He prepares himself a sandwich, sits down on the couch and switches on the TV set. He starts watching "rbb AKTUELL". The first spots are mainly about politics and about Berlin. After a while there is the first really interesting news for Ralph: Soccer News! It is not about sports though, actually, but about two professional kickers who admitted involvement in a series of robberies. The two former professional soccer players from third division club SV Babelsberg 03, Suleyman Koc and Guido Kocer, have to answer the Berlin District Court since Tuesday. Ralph is especially interested, because he follows this local club and one of his friends even went to school with Kocer. First Ralph checks an article with the details of the robberies – he had almost forgotten them, because they were committed more than a year ago and then he hadn’t really listened. Now that he realised he (almost) knew one of the responsibles, his interest has grown immediately. After watching the spot himself, he shares it with his friend who went to school with Kocer. He sends him a recommendation to watch the next show of “Brandenburg aktuell” for which they just announced a longer interview with the club manager. Just in case, he also bookmarks it for himself, so he might watch it tonight. One of the next spots presents a marketing campaign of hotels in Berlin: “Experience your city". At the beginning of next year, many hotels invite locals to enjoy a Berlin weekend from a tourist's perspective: people from the region can stay in a double room of a premium hotel for only 99 € for a whole weekend. Ralph likes the whole idea and follows a link to the campaign’s website. He thinks it would be a nice present for his girlfriend Cindy. He "likes" this news item and receives a notification that his (social network) friends Holger and Janine had also liked it. Holger is online, so Ralph starts chatting with him. They have the idea to invite some friends to book rooms for the same hotel and the same night, so they could celebrate Cindy's birthday together. The door bell rings and his girlfriend Cindy comes in. Ralph invites her to watch the rest of the show together. Next up is a spot about the restoration of a church at a nearby lake; as a carpenter, Ralph is always interested in the restoration of old buildings. As Cindy would not be interested, he stores a bookmark to dig in deeper when there is time and quickly skips the item. While Ralph would love to watch the next spot on another famous church in Berlin, Cindy would prefer to browse through those spots she had missed. Rather than watching different spots side by side, Ralph skips to the weather forecast, and invites Cindy for a walk. Maybe he can find out, if she would be interested in a hotel weekend in Berlin before he confirms the booking.

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Name and occupation: Ralph, carpenter	**Carpenter** as instanceOf Occupation		http://dbpedia.org/resource/carpenter	5
Nationality / place of residence: German / Prenzlau (Brandenburg)	**Germany** as instanceOf Country, **Prenzlau** as instanceOf City, **Brandenburg** as instanceOf State		http://dbpedia.org/resource/Germany http://dbpedia.org/resource/Prenzlau http://dbpedia.org/resource/Brandenburg	5 5 5
Main interests: Prenzlau and surroundings, architecture, nature, joinery, crafts, local sport clubs, boats, cars, car mechanics	**Joinery** as instance of Technology	**Architecture** as subClassOf Arts, **Nature** as subClassOf Topic, **Crafts** as subClassOf	http://dbpedia.org/resource/Category:Architecture http://dbpedia.org/resource/Category:Nature http://dbpedia.org/resource/Category:Joinery http://dbpedia.org/resource/Category:Craft http://dbpedia.org/resource/Category:Boat http://dbpedia.org/resource/Category:Car	10 10 5 5 5 5
After school he served an apprenticeship as carpenter in the neighbouring village Strehlow.				
Both spend much time together and like to go for a walk. In summer they often go swimming in the nearby lake.		**Swimming** as subClassOf Sports	http://linkedtv.eu/Topic/Sports/Swimming	10

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Then Ralph is looking at the boats too, often thinking it would be nice to have one too one day, but he will have to learn to sail first.		**Sailing** as subClassOf Sports	http://linkedtv.eu/Topic/Sports/Sailing	10
They love nature and look forward to moving into their own house near the forest one day.				
Being a carpenter, he is very interested in materials, building fabric, constructions and architecture. On the other hand, he is very interested in all local sports clubs.			http://dbpedia.org/resource/Category:Wood http://dbpedia.org/resource/Category:Construction http://dbpedia.org/resource/Category:Building_materials	5 5 5

Nina / Urban mom

- Name and occupation: Nina, teacher on maternal leave (Mother of Lisa)
- Age: 32
- Nationality / place of residence: German / Prenzlauer Berg
- Digital literacy: interested in new media and “hip” technology
- Main interests: Berlin, city life, theatre, education, politics, music, culture, food, books, Pilates, cooking

Nina is a typical inhabitant of Berlin’s hippest quarter, Prenzlauer Berg. She is a well educated and well informed young mother. She really likes discussing things. She especially likes to talk about politics and culture. Therefore, she likes to get deeper and not just superficial information. When a subject is interesting for her, she will take the time to understand it properly. She uses media in a very organized manner and especially about watching TV she is very picky. She likes Berlin with the constant changes and she feels very much at home in her family-friendly neighbourhood. She likes to visit exhibitions and also to go to the theatre and readings. Because she is very interested in culture, she would like to be informed about the city life, current events and new galleries. She does not like it at all when things are complicated and take a long time to understand and she can get impatient very quickly, so every application or service has to be smooth and easy. Nina is not interested in technological background information, on how a system works, it should just work well and adapt to her life; time-independence is also very important for her, because of her little girl: she only has time to watch infotainment programmes whenever her daughter is asleep.

Nina's baby has fallen asleep after feeding, so Nina switches on the TV to be informed while doing some housework. Browsing the programme she sees that yesterday's enhanced "rbb AKTUELL" evening edition is available and starts the programme. Nina watches the intro with the headlines while starting her housework session with ironing some shirts. Watching a news spot about Berlin's Green Party leader, Volker Ratzmann, who withdrew from his office yesterday, Nina is kind of frustrated as she voted for him and feels her vote is now "used" by someone she might not have voted for. She would like to hear what other politicians and people who voted for him think about Ratzmann's decision to resign. She watches a selection of video statements of politicians and voters. What she loves most about this is that she can sort these statements according to position (professional commentary vs. street poll) or political background (Green, conservative,..). Watching a news spot about a debate on the mandatory labelling for police officers, Nina asks herself why the labelling of policemen should be changed. Out of interest, she browses through a list of videos from the debate and chooses a speech by Benedikt Lux who seems very interesting to her. During his statement she reads some additional information about him: since when does he represent the Green party in Berlin’s senate, what are his political fields of interest, etc. Nina finally gives up the idea of ironing and prepares her stuff for a walk with the stroller and for the toddler group after that. Nina keeps watching some news items from the corner of her eye, but doesn't really pay attention anymore. A spot about Wolf Biermann’s 75th birthday sounds really interesting, so she bookmarks it. Maybe she can watch it later when the baby is sleeping.

She decides to quickly skip to the weather forecast. Oh, rainy weather in the afternoon. She grabs a raincoat, her child and the stroller, switches off the television and leaves the house.

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Nationality / place of residence: German / Prenzlauer Berg	**Germany** as instanceOf Country, **Berlin-Prenzlauer Berg** as instanceOf City, **Berlin** as insanceOf City		http://dbpedia.org/resource/Germany http://dbpedia.org/resource/Prenzlauer_Berg http://dbpedia.org/resource/Berlin	5 5 5
interested in new media and "hip" technology		**NewMedia** as subClassOf MediaType	http://dbpedia.org/resource/Category:New_media	10
Main interests: Berlin, city life, theatre, education, politics, music, culture, food, books, Pilates, cooking	**Pilates** as instanceOf Gymnastics	**Theatre** as subClassOf Arts, **Education** as subClassOF Society, **Politics** as subClassOf Society, **Culture** as subClassOf Topic, **CityLife** as subClassOf Culture, **Music** as subClassOf Topic, **Food** as subclassOf Health, **Cooking** as SubClassOf Hobbies, **Books** as subClassOf MediaType	http://dbpedia.org/resource/Theatre http://dbpedia.org/resource/Category:Education http://dbpedia.org/resource/Category:Politics http://dbpedia.org/resource/Category:Culture http://dbpedia.org/resource/City_Life http://dbpedia.org/resource/Category:Music http://dbpedia.org/resource/Category:Food http://dbpedia.org/resource/Cooking http://dbpedia.org/resource/Category:Book http://dbpedia.org/resource/Pilates	10 10 10 10 10 10 10 10 10 5
She likes to visit exhibitions and also to go to the theatre and readings.		**Exhibition** as subClassOf Arts, **Readings** as subClassOf Arts	http://dbpedia.org/resource/Category:Exhibition http://dbpedia.org/resource/Reading	10 10

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Because she is very interested in culture, she would like to be informed about the city life, current events and new galleries.		**Event** as subClassOf Thing, **Gallery** as instanceOf Exhibition	http://linkedtv.eu/Event http://dbpedia.org/resource/Gallery	10 10
She only has time to watch infotainment programmes whenever her daughter is asleep.		**Information** as subClassOf TV, **Entertainment** as subClassOf TV	http://linkedtv.eu/Topic/TV/Information http://linkedtv.eu/Topic/TV/Entertainment	10 10

Peter / Socially active retiree

- Name and occupation: Peter, retired widower
- Age: 65
- Nationality / place of residence: German / Potsdam
- Digital literacy: early adopter; interested in every new technology
- Main interests: Brandenburg and especially Potsdam, evening events for seniors in neighborhood, technology, television, gardening, sailing, health, medicine, travelling

Peter lives in Potsdam a small and wealthy town near Berlin. Since his retirement he has a lot of time for his hobbies. He likes watching TV and listening to the radio. He is also very interested in new technology and likes to use new services via the internet. He especially appreciates how easy new technology makes staying in contact with his family living further away. He never hesitates to become acquainted with new technology and operating concepts and if he likes something he optimizes its handling. He has certain favourite TV programs and is particularly interested in the news from his region. He is involved in several activities in his neighbourhood and likes to talk about his views. For example, he talks with the bakery saleswoman about the upcoming public festival in Potsdam or about regional politics. Since Peter consumes a lot of media throughout the day, it often happens that news is repeated. On the other hand, there are often topics he wants to know more about than what is included in the usual news report. In addition to regional politics and events he is also interested in gardening, travelling, new technology trends as well as health and medical issues.

After a long walk in the sun around the lake alone, Peter feels refreshed and thirsty for information. His favourite source is the local news magazine "rbb AKTUELL". He gets himself some coffee from the kitchen and sits down to see what happened while he was out and about. In the main news of the day, Peter hears that Berlin’s Green party leader threw in the towel yesterday. He is not particularly interested in Berlin’s politics, but this guy seemed to be quite smart from what Peter understood in the short excerpt of the interview. Peter would like to know more about Ratzmann and why he decided to leave the political stage, so he switches to the longer version of this interview. One of the next spots is about a fire at famous Café Keese in Berlin. Peter is shocked. He used to go there several times, but that was years ago. He wonders how the place may have changed over the years. In the news spot, smoke and fire engines was almost all one could see, so he checks some older videos about the story of the famous location where men would call women on their table phones – hard to believe nowadays, now that everyone carries around mobile phones! Memories of these good old days make him happy and sad at the same time. After checking these very nice clips, he returns to "rbb AKTUELL" and watches the next spot on a new Internet portal about rehabilitation centres in Berlin and Brandenburg. He knows an increasing number of people who needed such facilities. He takes a look at a map of Brandenburg showing the locations of these centres and bookmarks the linked portal website to check some more information later. At the end of the show, he takes an interested look at the weather forecast, hoping that tomorrow would be as nice as today so he could go out again to bask in the sun.

User profile description	Automatically inferred interest	Explicit selected interest	URI link	Weight
Nationality / place of residence: German / Potsdam	**Germany** as instanceOf Country, **Potsdam** as instanceOf City		http://dbpedia.org/resource/Germany http://dbpedia.org/resource/Potsdam	5 5
Digital literacy: early adopter; interested in every new technology		**Technology** as subClassOf Topic	http://dbpedia.org/resource/Category:Technology	10
Main interests: Brandenburg and especially Potsdam, evening events for seniors in neighborhood, technology, television, gardening, sailing, health, medicine, travelling	**Brandenburg** as instanceOf State	**EveningEvent** as instanceOf SocialEvent, **TV** as subClassOf Topic, **Gardening** as subClassOf Hobbies, **Sailing** as subClassOf Sports, **Health** as subClassOf Topic, **Medicine** as similarTo MedicalEntity, **Travelling** as subClassOf Hobbies	http://dbpedia.org/resource/Brandenburg http://schema.org/SocialEvent http://linkedTV.eu/Topic/TV http://linkedtv.eu/Topic/Hobbies/Gardening http://linkedtv.eu/Topic/Sports/Sailing http://linkedtv.eu/Topic/Hobbies/Travelling http://linkedtv.eu/Topic/Health http://linkedtv.eu/Topic/Health/MedicalEntity	5 10 10 10 10 10 10 10
He likes watching TV and listening to the radio.		**Radio** as subClassOf ??	http://dbpedia.org/resource/Radio	10
He is also very interested in new technology and likes to use new services via the internet.		**Internet** as subClassOf Technology	http://linkedtv.eu/Topic/Technology/Internet	10
He has certain favourite TV programs and is particularly interested in the news from his region.		**Entertainment** as subClassOf TV, **News** as subClassOf TV	http://linkedtv.eu/Topic/TV/Entertainment http://linkedtv.eu/Topic/TV/News	10 10
For example, he talks with the bakery saleswoman about the upcoming public festival in Potsdam or about regional politics.		**SocialEvent** as subClassOf Event, **Politics** as subClassOf Society	http://linkedtv.eu/Event/SocialEvent http://linkedtv.eu/Topic/Society/Politics	10 10

14.2 Appendix-2: f-pocketKRHyper supported semantics

The supported semantics of concepts (primitive and complex), roles (object properties) for f-pocketKRHyper are described in the following tables. Let *C* and *D* be concepts, *R* and *P* be roles (object properties), *a* and *b* be individuals.

For the fuzzy semantics description we will use $\bowtie \in \{\geq, >, \leq, <\}$, $\lhd \in \{\geq, >\}$, $\rhd \in \{\leq, <\}$ and also $\otimes$ to symbolize the t-norm (= intersection in fuzzy set theory = conjunction in fuzzy logic), $\oplus$ to symbolize the t-conorm (= union in fuzzy set theory = disjunction in fuzzy logic) and $\ominus$ to symbolize the negation (= complement) [BOB12]. Let *d* and *d'* $\in [0, 1]$ be membership degrees of two concept assertions and $w \in [0, 1]$ the weight that modifies a membership degree.

Table 2: Class/concept & role constructors, axioms and facts supported

Abstract definition	OWL syntax	DL syntax	Crisp semantics
top	owl:Thing	$\top$	Δ^I
bottom	owl:Nothing	$\bot$	$\varnothing$
primitive concept	A (URI reference)	A	$A^I \subseteq \Delta^I$
concept negation	complementOf	$\neg C$	$\Delta^I \setminus C^I$
concept implication	subclassOf	$C \sqsubseteq D$	$C^I \subseteq D^I$
concept equivalence	equivalentClass	$C \equiv D$	$C^I = D^I \quad (C^I \subseteq D^I, D^I \subseteq C^I)$
concept conjunction	intersectionOf(*C*,*D*)	$C \sqcap D$	$C^I \cap D^I$
concept disjunction	unionOf(*C*,*D*)	$C \sqcup D$	$C^I \cup D^I$
concept disjointness	disjointWith	$C \sqcap D \sqsubseteq \bot$	$C^I \cap D^I \subseteq \varnothing$
(non-symmetrical) existential quantification	(*R* someValuesFrom(*C*)) subclassOf *D*	$\exists R.C \sqsubseteq D$	$\{x \mid \exists y\ (x,y) \in R^I \cap y \in C^I \rightarrow x \in D^I\}$

(non-symmetrical) universal quantification	subclassOf C (R allValuesFrom(D))	$C \sqsubseteq \forall R.D$	$\{x \in C^{I} \rightarrow (x \mid \forall y\ (x,y) \in R^{I} \rightarrow y \in D^{I})\}$
role	R (URI reference)	R	$R^{I} \subseteq \Delta^{I} \times \Delta^{I}$
role implication	subPropertyOf	$R \sqsubseteq P$	$\{\forall x,y\ (x,y) \in R^{I} \rightarrow (x,y) \in P^{I}\}$
role equivalence	equivalentProperty	$R \equiv P$	$R^{I} = P^{I}$ ($R^{I} \subseteq P^{I}$, $P^{I} \subseteq R^{I}$)
inverse role	inverseOf	$R^{-} \equiv P$	$\{\forall x,y\ (x,y) \in R^{I} \leftrightarrow (y,x) \in P^{I}\}$
transitive role	transitiveProperty	R^{+}	$\{\forall x,y,z\ (x,y) \in R^{I} \cap (y,z) \in R^{I} \rightarrow (x,z) \in R^{I}\}$
individual	a (URI reference)	a	$a^{I} \in \Delta^{I}$

Table 3: Fuzzy semantics and the degree determination rules given an existing assertion for each concept in all depicted axioms

Abstract definition	**Syntax**	**Semantics**
fuzzy concept assertion	$\langle a : C \triangleright\triangleleft d \rangle$	$C^{I}(a^{I}) \bowtie d$
weighted concept	$w \cdot C$	$C^{I}(a^{I}) \bowtie w \cdot d$
Negation	$\neg C$	$\ominus C^{I}(a) \bowtie d$
implication	$C \sqsubseteq D$	$C^{I}(a^{I}) \bowtie d \rightarrow D^{I}(a^{I}) \bowtie d$
conjunction	$C \sqcap D$	$C^{I}(a^{I}) \bowtie d \otimes D^{I}(b^{I}) \bowtie d'$
disjunction	$C \sqcup D$	$C^{I}(a^{I}) \bowtie d \oplus D^{I}(b^{I}) \bowtie d'$
quantification	$\exists R.C \sqsubseteq D$	$R^{I}(a^{I},b^{I}) \cap C^{I}(b^{I}) \bowtie d \rightarrow D^{I}(a^{I}) \bowtie d$
	$C \sqsubseteq \forall R.D$	$C^{I}(a^{I}) \bowtie d \rightarrow R^{I}(a^{I},b^{I}) \rightarrow D^{I}(b^{I}) \bowtie d$

Table 4: Zadeh fuzzy operations

⊗	⊕	⊖
min(⋈d , ⋈d')	*max(⋈d, ⋈d')*	¬⋈(1−*d*)

Table 5: The negation ¬⋈of an operator ⋈ [BOB12]

⋈	¬⋈
≥	<
>	≤
≤	>
<	≥

