AVATAR: An Advanced Multi-agent Recommender System of Personalized TV Contents by Semantic Reasoning

Yolanda Blanco-Fernández, José J. Pazos-Arias, Alberto Gil-Solla, Manuel Ramos-Cabrer, Belén Barragáns-Martínez, Martín López-Nores, Jorge García-Duque, Ana Fernández-Vilas, and Rebeca P. Díaz-Redondo

Department of Telematic Engineering, University of Vigo, 36310, Spain
{yolanda,jose,agil,mramos,belen,mlnores,jgd,avilas,rebeca}@det.uvigo.es

Abstract. In this paper a recommender system of personalized TV contents, named AVATAR[1], is presented. We propose a modular multi-agent architecture for the system, whose main novelty is the semantic reasoning about user preferences and historical logs, to improve the traditional syntactic content search. Our approach uses Semantic Web technologies – more specifically an OWL ontology – and the TV-Anytime standard to describe the TV contents. To reason about the ontology, we have defined a query language, named LIKO, for inferring knowledge from properties contained in it. In addition, we show an example of a semantic recommendation by means of some LIKO operators.

1 Introduction

Nowadays, a migration from analogue to digital TV is taking place in TV. This change has two main implications: the capacity to broadcast more channels in the same bandwidth, and the possibility to send software applications mixed with audiovisual contents. The TV recommenders play a key role in this scenario because they can help the users to find interesting contents among a large amount of irrelevant information.

Several different approaches have appeared in the field of TV recommender systems, such as Bayesian techniques [7], content-based methods [8], collaborative filtering [9], decision trees [10]. In this paper, a new recommender system is presented, named AVATAR, that combines different strategies to improve the success of recommendations. Among others, we use Bayesian techniques, semantic reasoning and profiles matching. Here the semantic reasoning approach is described, as we think it is a novel and promising method to enhance the elaborated suggestions by our system.

Such a reasoning process requires a high degree of normalization. In this regard, we use the TV-Anytime standard (*www.tv-anytime.org*), which normalizes descriptions of generic TV contents, concrete instances of programs and user profiles. On the other hand, our personalization tool needs a knowledge base to feed the reasoning process. This work extends the use of the Semantic Web [6] technologies to the TV context. So, we have implemented an ontology about TV contents according to the OWL language.

[1] Work supported by the Ministerio de Educación y Ciencia Research Project TSI2004-03677.

X. Zhou et al. (Eds.): WISE 2004, LNCS 3306, pp. 415–421, 2004.

Finally, note that we conceive our system as a MHP interactive application, downloaded from the service provider through the transport stream, achieving a wide deployment.

This paper is organized as follows. Sect. 2 describes the architecture of the TV recommender proposed. In Sect. 3, we focus on different issues referred to the semantic reasoning and in Sect. 4 we show how it can be used in the context of personalized TV through an example. Finally, we present some conclusions and discuss future work.

2 The Architecture of the AVATAR Recommendation Service

In this section, the main design decisions of the AVATAR architecture are presented. We propose an open and modular architecture that allows to update the modules that generate recommendations and to add new ones that compute suggestions by other strategies.

2.1 The AVATAR Recommender System: A MHP Interactive Application

As commented in the introduction, the system must be flexible enough to be updated frequently. For this reason, we conceive our system as a MHP interactive application.

In the DVB MHP standard, applications are executed in the context of concrete services or events in a service, and, usually, they do not survive after finishing that context (*event final* or *service change*). Taking into account that AVATAR needs to record all the viewers actions beyond a concrete service, our approach integrates a special agent, named *local agent*, to know the user viewing behavior all the time.

Our prototype uses the TV-Anytime standard to store the historic logs of the users and their personal preferences about TV contents. The real format of the data stored by the *local agent* might be private and in this case, the procedure of access the information must be normalized. So, we propose a new MHP API (TV-Anytime MHP API) to provide a neutral way to access information described with TV-Anytime metadata, even though the local agent does not use this format to store these data.

2.2 AVATAR: A Modular Multi-agent Recommender System

As we said previously, we propose a modular architecture, in which the recommender is divided into two parts. The first one is related to the local software of the STB, that requests the personal data and user preferences, and records information about the TV contents already watched. This information would be accessible for the recommender through the TV-Anytime MHP API. For the second part, the MHP application implements the functionality of the recommendations service. It consists of three modules:

Recommenders. They are agents that implement the different strategies to make personalized recommendations. Fig. 1 shows three of them: an agent based on Bayesian techniques, another one based on semantic reasoning and the last one, based on profiles matching. Their recommendations are mixed by the combiner module, that is a neural network [1]. This recommendation is stored as private data and compared with the user choices to improve future suggestions. The semantic agents need a knowledge base to reason about TV contents and user preferences. The knowledge base in our system is

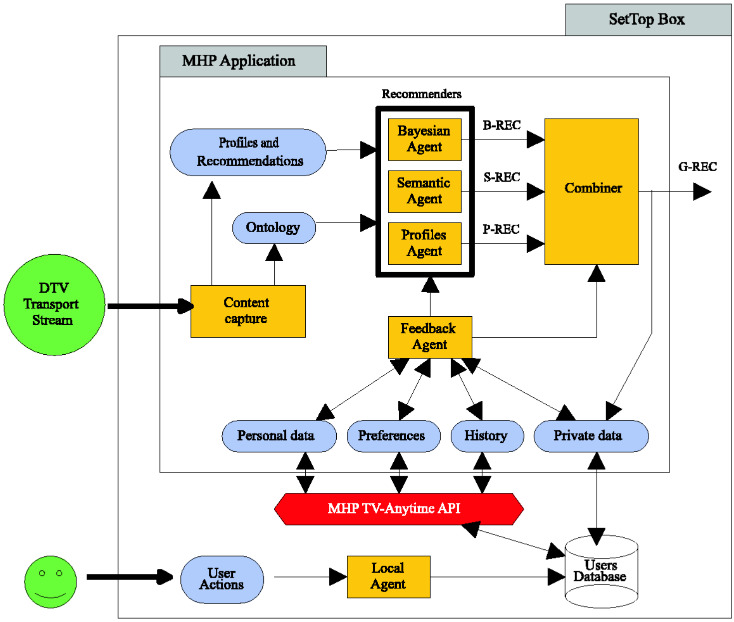

Fig. 1. The architecture of a broadcast recommender system

an OWL ontology (*http://avatar.det.uvigo.es/ontology*) that allows to share and reuse information efficiently in a multi-agent architecture [5].

Capture and classification of information. The goal of this module is the capture of the TV contents, described by TV-Anytime metadata, and their classification into the appropriate ontological classes. Besides, the tool receives a set of prototypical profiles together with recommendations for each one of them in order to reuse suggestions previously made for viewers with analogous profiles.

The feedback system. The feedback agent accesses to the feedback information stored in a user database by the TV-Anytime MHP API, updates the user profiles and feeds the recommender agents with information for the inference process.

The modular character of the architecture proposed ensures its openness. Thus, only small changes would be necessary to add new strategies to make recommendations.

3 The Semantic Reasoning Approach

The Semantic Web is a novel approach that intends to get the computers can "understand" the handled information [6]. Just like it happens in TV recommender systems, one of the main goals is to attain a high degree of personalization in the services offered to the Web users. On the other hand, the Semantic Web thinks of the Web as a repository of knowledge [2], so that there exist mechanisms to share and manage the knowledge efficiently, such as ontologies [4]. For that reasons, we propose the application of Semantic Web technologies in the domain of TV.

Table 1. The query operators of the LIKO language

Operator	*Input Parameter*	*Description*
$\triangleright$	A superclass that can also be a subclass in the ontology.	Returns a set of subclasses whose superclass is indicated in the field parameter.
$\triangleleft$	A subclass that can also be a superclass in the ontology.	Returns a set of superclasses of the subclass indicated as input parameter.
$>>$	A superproperty defined in the ontology.	Returns the subproperties of the indicated property.
$<<$	A subproperty defined in the ontology.	Returns a set of superproperties of the indicated property.
$>\Leftarrow$	Instances of classes of the ontology.	Returns the properties where the domain contains the indicated class.
$>\Rightarrow$	Instances of classes of the ontology and values of Datatype properties.	Returns a set of properties where the range contains the indicated parameter.
$>\odot$	Instances of classes or Datatype properties values.	Returns properties inferred from the transitive properties of the ontology.
$>\oplus$	Instances of classes or Datatype properties values.	Returns properties inferred from the functional properties in the ontology.
$>\ominus$	Instances of classes or Datatype properties values.	Returns the properties inferred from the inverse functional properties.
$>\hookleftarrow$	Instances of classes or datatype properties values.	Returns a set of properties inferred from symmetric properties of the ontology.

So, the OWL ontology that we have implemented is a tree with several levels. The root node is named “TV Contents”. The second level consists of general programs (such as “Movies” and “Documentaries”), the third level are more specialized contents (i.e. “Action Movies” and “Nature Documentaries”) and so on. The reasoning process is carried out in two phases. Firstly, AVATAR must choose a set of general contents (programs in the tree second level) according to the user profile. For that purpose, AVATAR uses a query language, named *LIKO* (see Table 1) , to reason about classes and properties contained in the TV ontology. Finally, the tool must select the appropriate instances of the chosen classes to enhance the offered recommendations.

These operators can be combined by unions or intersections. The operators that infer knowledge from transitive, functional, inverse functional and symmetric properties are implemented by the rest of *LIKO* operators. It is obvious that to infer data with the $>\odot$ operator by the ontological transitive properties, is necessary to analyze the properties range and domain (through $>\Rightarrow$ and $>\Leftarrow$, respectively), to make reasonings such as *IF* $a \rightarrow b$ *and* $b \rightarrow c$ *THEN* $a \rightarrow c$. A use example of the $>\odot$ operator is shown in Sect. 4.

4 An Example of Recommendation Based on Semantic Reasoning

In this section, we show how the AVATAR system applies the *LIKO* query operators about some properties contained in the ontology.

Assume that a user registers in AVATAR. He is married and works as a doctor. Besides, he has watched “Safaris in Kenya” (a documentary) and “CNN News”.

Table 2. Applied operators to reason about the user's job

Query operators	*Obtained properties and classes*
Doctor $>\Rightarrow \cup$ Doctor $>\Leftarrow$	Doctor $\leftarrow$ WorksIn $\rightarrow$ Hospitals
Hospitals $>\Rightarrow \cup$ Hospitals $>\Leftarrow$	IncidentsNews $\leftarrow$ hasSynopsis $\rightarrow$ "Hospitals strike in Madrid"
Incidents News $\lhd$	News
News $\lhd$	Informative Programs

Table 3. Applied operators to reason about the user's marital status

Query operators	*Obtained properties and classes*
Married People $>\Rightarrow \cup$ Married People $>\Leftarrow$	Married People $\leftarrow$ interestedIn $\rightarrow$ Cruises
Cruises $\lhd$	Travels
Travels $\lhd$	Advertising Products
Advertising Products $\lhd$	NULL
Cruises $>\Rightarrow \cup$ Cruises $>\Leftarrow$	Advertising Product $\leftarrow$ hasDescription $\rightarrow$ "Cruise on the Nile" Advertising Product $\leftarrow$ hasDescription $\rightarrow$ "Cruise on the Caribbean"
Advertising Product $>\Rightarrow \cup$ Advertising Products $>\Leftarrow$	Advertising $\leftarrow$ hasProducts $\rightarrow$ Advertising Products

Firstly, AVATAR reasons about the user's job as seen in Table 2. The first two *LIKO* operators relates the user's working place to the "Incidents News" class. Next, AVATAR locates one personalized content of interest for the user because it is related to his job.

AVATAR also reasons about the user marital status. So, as seen in Table 3, the first operator finds that the cruises are appealing to the married people. Later, AVATAR must choose an appropriate region for the cruise. Next, the system finds out the superclass referred to the "Cruises" class, until to get the "Advertising Products" superclass. When the reasoning by the search of superclasses is not possible, AVATAR uses other operator about "Cruises". So, the system establishes relations to the "Advertising" category.

Once the "Advertising" and "Informative Programs" classes have been chosen, the tool will not consider all the ontological categories in the following reasoning phase, reducing greatly the complexity of the inference process. Next, the system uses information about the user's view history so as to find relations between the watched programs and the TV contents computed previously. Remember that the user view history contains news and documentaries. AVATAR allows to add brief personalized subtitles during the viewing of the programs selected by the user, related to news of interest. So, AVATAR recommends subtitles referred to a strike in hospitals.

AVATAR continues reasoning by means of the datatype property *"Safaris"* $\leftarrow$ *hasDescription* $\rightarrow$ *"Safaris in Kenya."*. Next, AVATAR uses the $>\odot$ operator as shown in Table 4 to reason about the transitive property "isIn" by exploring its range and domain.

Table 4. An example of reasoning involving the $>\odot$ query operator

Query operators	*Obtained properties*	*Inferred knowledge*
Kenya $>\Rightarrow$	NULL	
Kenya $>\Leftarrow$	Kenya $\leftarrow$ isIn $\rightarrow$ Africa	
Africa $>\Rightarrow$	Egypt $\leftarrow$ isIn $\rightarrow$ Africa	
Africa $>\Leftarrow$	NULL	
Egypt $>\Rightarrow$	Nile $\leftarrow$ isIn $\rightarrow$ Egypt	Nile $\leftarrow$ isIn $\rightarrow$ Africa and Kenya $\leftarrow$ isIn $\rightarrow$ Africa

After applying the $>\odot$ query operator, the system has discovered a common nexus between Kenya and the Nile: both regions are in Africa. This way, AVATAR can include in the informative subtitles previously described news about incidents happened in Africa. Remember that AVATAR had found two interesting commercials about cruises (see Table 3). The relation between the Nile and Africa allows to choice an appropriate region for this cruise. Taking into account that the user had watched a documentary about Kenya, a region in Africa, it is most appropriate to suggest a cruise on the Nile.

In a real scenario, we should need a knowledge base with a large amount of data, to obtain recommendations by means of discovering relations between users personal data and information contained in the TV ontology. For that reason, we are focusing on the implementation of a LIKO-based semantic matching algorithm to find out semantic associations among the users favourite programs and other TV contents.

5 Conclusions and Further Work

In this paper we have presented a TV recommender system, named AVATAR, conceived as a MHP application. For this system, we have described an open multi-agent architecture that allows to include modules with additional functionalities easily to enhance the made recommendations. Our approach is novelty so that improves the previous TV recommendation tools by incorporating reasoning capabilities about the content semantics. For that purpose, we have used the TV-Anytime initiative to describe the TV contents and a TV ontology to share and reuse the knowledge efficiently [3]. To discover semantic associations among different TV contents, the system employs a query language that infers data from the AVATAR knowledge base.

Our future work is related to the application of the Description Logics to improve the semantic reasoning by incorporating new inference rules into our knowledge base.

References

1. Ardissono L. et al. *Personalized Digital TV: Targeting Programs to Individual Viewers*, chapter 5. 2004.
2. Daconta M. et al. *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*. John Wiley & Sons, 2003.
3. Davies J. et al. OntoShare: Using Ontologies for Knowledge Sharing. In *Proceedings of the 11th International WWW Conference WWW2002*, 2002.
4. Davies J. et al. *Towards the Semantic Web: Ontology-driven Knowledge Management*. John Wiley & Sons, 2003.

5. Decker S. et al. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In *DS-8*, pages 351 – 369, 1999.
6. Geroimenko V. et al. *Visualizing the Semantic Web*. Springer Verlag, 2003.
7. Heckerman D. Bayesian networks for data mining. *Data Mining and Knowledge Discovery*, pages 79–119, 1997.
8. Mouaddib A. et al. Knowledge-based anytime computation. In *Proceedings of IJCAI*, 1995.
9. Resnick P. et al. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of Conference on Computer Supported Cooperative Work*, 1994.
10. Sekine S. et al. A decision tree method for finding and classifying names in japanese texts. In *Proceedings of the Sixth Workshop on Very Large Corpora*, 1998.