AVATAR: A Flexible Approach to Improve the Personalized TV by Semantic Inference ⋆

Yolanda Blanco Fernández, Jose J. Pazos Arias, Alberto Gil Solla and Manuel Ramos Cabrer

Department of Telematic Engineering, University of Vigo, 36310, Spain

Abstract. Both the TV recommender systems and search engines developed in the Internet are intended to lighten the user burden, by offering them automatically the required information, personalized according to their preferences or needs. In last years, with the goal of improving these search engines, an important research line has been developed in the context of the WWW, known as the Semantic Web. The Semantic Web describes the resources by metadata and reasons on them by discovering new knowledge. Taking the advantage of the Semantic Web in the field of the personalized TV, we propose an intelligent assistant named AVATAR, which uses the semantic inference as a novel recommendation strategy. This approach allows to overcome an important limitation identified in the personalization strategies adopted in other systems: an excessive similarity between the programs known by the user and those suggested by the recommender. In this regard, our approach diversifies and personalizes the elaborated recommendations, by inferring semantic associations of different nature between the user preferences and the suggested TV contents. This inference process requires a formal representation both the knowledge of our application domain, and the user preferences. In this regard, we resort to an OWL ontology to identify resources and relations typical in the TV field, and to reason about them.

1 Introduction

The adoption of different standards for Digital TV (DTV) envisages a scenario in which the users can access to a greater number of audiovisual contents, transmitted together with interactive applications. This situation causes the viewers feel disoriented among a massive amount of irrelevant information. To address effectively this problem it is necessary to resort to tools —named *TV recommender systems* [1]— which lighten the user burden by offering automatically programs personalized according to their preferences.

The functionality of these recommenders is very similar to the goal pursued by the search engines developed in the Web, whose utility is undeniable due to the overwhelming information available for their users. In last years, the so-called Semantic Webhas become a relevant research line. The Semantic Web uses metadata for describing the Web resources, so machines can understand these descriptions and infer semantic relationships between them. Consequently, the search engines for the Semantic Web overcome clearly the simple syntactic approach adopted in tools like Google.

⋆ Work supported by the *Ministerio de Educación y Ciencia* Research Project TSI2004-03677

Blanco Fernández Y., J. Pazos Arias J., Gil Solla A. and Ramos Cabrer M. (2005).
AVATAR: A Flexible Approach to Improve the Personalized TV by Semantic Inference.
In *Proceedings of the 1st International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces*, pages 69-78
DOI: 10.5220/0001421100690078

Taking the advantage of the Semantic Web, the TV recommenders can also be improved by including capabilities of semantic inference. This way, here we present AVATAR (**A**d**VA**nced **T**elematic search of **A**udiovisual contents by semantic **R**easoning), a TV recommender which suggests to a user contents semantically related to those he/she watched in the past. This novel recommendation strategy enhances the offered suggestions by discovering appealing relationships, unknown in previous approaches. Such relationships —named *semantic associations*— are inferred from the knowledge learned by our intelligent assistant.

Such inference process permits to overcome a drawback identified in previous recommendation approaches, that is, the excessive similarity between the contents known by the user and those suggested by the system. To diversify the elaborated suggestions, our proposal considers a complete classification of semantic associations of diverse nature. In this regard, the kind of discovered association and the diversification are closely related in our approach: the more direct the inferred relationships between the user preferences and the suggested programs, the less diversified this recommendation. Conversely, the discovery of indirect associations —not supported in the existing approaches— allow to suggest varied TV contents. It is worth noting that the diversification introduced in our system by the semantic inference is especially useful in the TV domain, where neither viewing habits of the users nor their preferences follow a homogeneous pattern. Consequently, the different types of associations permit to adjust the offered recommendation to each viewer, leading to successful personalized suggestions.

To carry out such inference process our system requires a mechanism to represent formally the knowledge of our specific application domain. In the Semantic Web, one of the well-known methodologies intended for this purpose are the ontologies. So, we have implemented an ontology —using the OWL (*Web Ontology Language*) language— where resources and relations typical in the TV domain (categories of programs, genres, credits involved in them, etc.) have been identified by means of classes, properties and specific instances. This way, our approach focuses on the explicit properties defined in the TV ontology, with the goal of inferring implicit semantic associations between the instances stored in this knowledge base.

On the other hand, taking into account that the ontologies are intended to reuse efficiently the knowledge represented in them [2], our approach has also used the TV ontology to model the user preferences, defined in personal profiles.

This paper is organized as follows: Sect. 2 reviews the related work and highlights the main differences with the aims of our proposal. Sect. 3 presents the structure of our OWL ontology and defines the user profiles handled in AVATAR. Sect. 4 describes the recommendation strategy used in our tool, including a complete classification of the supported *semantic associations* and explanatory examples. Finally, Sect. 5 summarizes the main conclusions and discusses future work.

2 Related Work

The personalization of services, according to user preferences, can be located in the 80's and bound to Internet. Personalization intends to resolve the problems caused by the more and more available information, firstly in News groups and later in the WWW.

It is in this context where the three main categories of systems for contents personalization are identified. From the perspective of the information which is analyzed to elaborate recommendations, the approaches can be classified in: (i) *content-based methods* which compute personalized recommendations by comparing content representations of previously liked items, with descriptions of items still unknown to the user; (ii) *collaborative methods*, which intend to recommend contents which have been interesting for users with similar preferences; and (iii) *hybrid methods*, which combine both methods to meet their different advantages [3]. Some well-known recommenders are the content-based NewsWeeder [4] and WebWatcher [5]; the collaborative Phoaks [6] and GroupLens [7]; and the hybrid Fab [8].

We have explored personalization systems for programming guides in the TV domain. As in the Internet context, the final aim is to avoid users managing such a huge amount of information. The first ones of these systems were offered from Web sites, for instance PTV [9]. Users register themselves in the PTV server (Web site) and then they can access to personalized programming guides presented as HTML or WML pages. The system incorporates user profiles, content-based reasoning and collaborative methods to make recommendations. When registering a new user, the system creates a profile which stores preferences about programs, channels, genres, timetables, etc.

These systems are able to provide a good service, as it is the case of PTVPlus, successor of PTV. Today PTVPlus is working in UK and Ireland and it provides personalization services of about a hundred of TV channels to thousands of people. However, two main lacks can be identified: the way in which user information is gathered to elaborate the profile, and the inability to maintain historic logs about what programs have been watched by what users. So, in these systems, it is not possible to deploy implicit techniques for managing the above information. The only solution is to use an explicit technique which elaborates an initial profile –from the information entered at registration– and after that, the feedback information (watched programs, changes in preferences, etc) needed for the explicit interaction and collaboration of the user through specific Web pages. Despite the fact that good recommendations can be achieved by this explicit mechanism, it entails the user to do a lot of work. Unfortunately, the user is not often willing to fill up a tedious form about his preferences, and then to access again and again to the WWW server in order to mark the programs.

Several ingenious solutions have been proposed to solve the problem of storing the user viewing history, such as the GuideRemote system. This proposal is based on a universal remote control that, after being connected to a PC, is able to download from the appropriate website the personalized programming guide for the following seven days, being also able to maintain information about the TV programs previously watched by the user and, after that, to inform the server. However, a personalization system running in the user receiver should be the most logic solution because all the information about the user and TV interaction is always available.

Having this information in mind, in this kind of systems is possible to set up implicit learning techniques about user interests and behavior. Additionally, this information can be continuously improved by taking into account all decisions adopted by the user any time he interacts with the system. This allows a great reduction in the information that must be explicitly given by the user so, in this case, the first time the user turns on the system only must inform about a few characteristics to built a preliminary profile.

Therefore, it is possible to find recommender systems [10] based on a multiagent architecture running in the user receiver. In these proposals specialized agents collaborate to obtain and combine the information broadcast with the information available in the Internet about TV programming guides. Besides that, they are also able to learn the user preferences and behavior in an implicit way, which allows having better recommendations in future. Likewise, [11] uses three different sources to build the user profile: his implicit viewing history, his explicit preferences and his interaction with the system. In this kind of systems, Bayesian techniques or those based on decision trees are the most used to allow the implicit learning of profiles.

Our approach differs from these works, given that our proposal uses several mechanisms to represent the knowledge about the TV domain, taking the advantage of previous experience in the Semantic Web. In order to carry out inference processes, our system requires generic descriptions about TV programs, that is, metadata about these TV contents.

Our proposal is based on finding out semantic relations between TV programs, by reasoning about the content descriptions provided by their metadata. The issue of the inference of complex relations has been only addressed out of the TV domain. Specifically, [12] defines this kind of associations for national security applications over a RDF(S) set. Its goal is to discover relationships between two instances specified by the user. Our approach differs from it because AVATAR stems from the viewer preferences, and finds out which contents are semantically related to them in a significant way. In other words, [12] resorts to two instances and AVATAR considers only one and discovers the second one, that is, the set of suggested programs.

3 Background

3.1 The TV Ontology

In the Semantic Web, ontologies are well-known methodologies for representing the knowledge of a specific application domain. In our proposal we have also resorted to an ontology, implemented using the OWL language, to conceptualize the TV domain. This ontology, referred to as $\mathcal{O}$, consists of two parts: (i) A *knowledge base* ($\mathcal{KB}_{\mathcal{O}}$) which contains classes and properties hierarchically organized, and (ii) a *description base* ($\mathcal{DB}_{\mathcal{O}}$) containing a set of instances of these classes related by the properties contained in $\mathcal{KB}_{\mathcal{O}}$. These instances are TV programs of different categories, actors, directors involved in them, genres, etc.

To identify relations among different resources, the concept of *property sequences* [12] is used. A *property sequence* PS is defined as a finite set of properties [P_0, P_1, . . ., P_N], which joins different classes stored in the *knowledge base* of the ontology $\mathcal{O}$. This way, the length of a sequence is defined as the number of properties contained in it.

Given that this ontology contains specific instances of each class, it is also possible to define a property sequence instance, represented by lower case letters. So, ps is a chain of properties which join different instances present in $\mathcal{DB}_{\mathcal{O}}$. The first node of

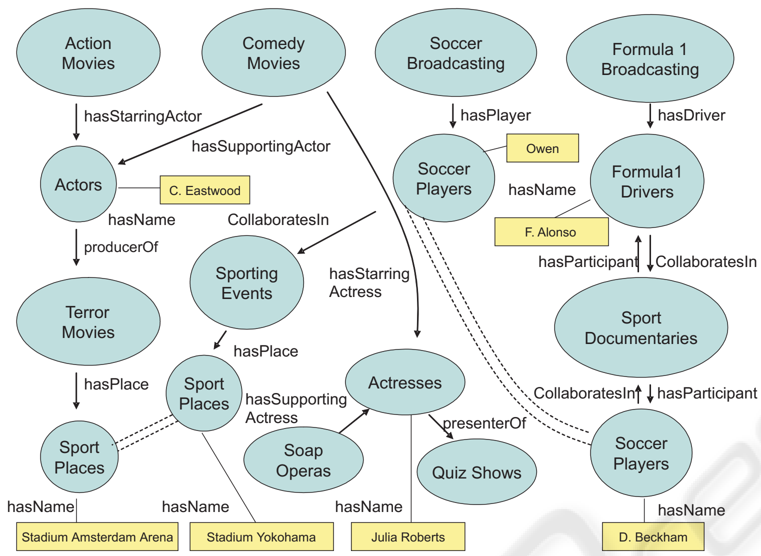

Fig. 1. Excerpt of $\mathcal{DB}_{\mathcal{O}}$

the instance ps is named *origin*, and the last one *terminus*. Besides, two functions have been used to return the classes contained in the sequence PS —`PS.NodesOfPS()`— and the instances contained in ps —`ps.PSNodesSequence()`.

Just as its name suggests, a property sequence PS (and an instance ps) fulfills that the class (instance) which appears as range in the property P_i, is the same one which figures as domain in P_{i+1}. To clarify the above concepts, an excerpt of our ontology is shown in Fig. 1, where the circles represent specific instances of classes with the same name, the arrows are properties, and the rectangles are concrete values of data types.

Here we can see different property sequences. For example, in the left side of the figure, the sequence ps = [hasStarringActor, producerOf, hasPlace, hasName] is shown. This property sequence allows to identify an action movie in which Clint Eastwood is involved as starring actor. Besides, this actor is also producer in a terror movie happened in the Stadium Amsterdam Area. This way, the nodes of this sequences are the following instances: `ps.PSNodesSequence()` = [Action Movies, Actors, Terror Movies, Sport Places], where *Action Movies* is the origin and *Sport Places* is the terminus of the sequence. On the other hand, the nodes contained in the set `PS.NodesOfPS()` would be the classes to which these instances belong. Note that in Fig. 1, we are representing instances and therefore, here *Action Movies* is a concrete instance of the class identified with the same name.

Next, we show how the knowledge represented in $\mathcal{O}$ is efficiently reused to model the user preferences.

3.2 The User Profiles

Each user profile defined in AVATAR stores personal information about the user and his/her preferences. To model these preferences we have used a dynamic subset of our

OWL ontology, built incrementally by adding new classes, properties and instances — extracted from $\mathcal{O}$— as the system knows additional information about his/her interests. For that reason, these profiles are named in AVATAR *ontology-profiles*. Specifically, when AVATAR knows a new content, the system adds to the profile: (i) this instance, (ii) the class referred to it, (iii) the hierarchy of superclasses defined in $\mathcal{KB}_{\mathcal{O}}$ related to this class and finally, (iv) some properties defining the main characteristics of this TV content.

For example, let us suppose that a user watched a Formula 1 race in which the driver Fernando Alonso is involved, and the action movie starring Clint Eastwood mentioned in the previous section. In this case, AVATAR adds to the user profile the instances *Action Movies* and *Formula1 Broadcasting* shown in Fig. 1, the classes to which these instances belong (they are classes with the same name than the mentioned instances), the superclasses *Sport Programs* and *Broadcasting Programs*, with respect to *Formula1 Broadcasting*, and the superclasses *Movies* with respect to *Action Movies*. Finally, some properties are also included in the user *ontology-profile*, such as *hasStarringActor*, *hasDriver*, etc, together with their respective values.

It is worth noting that the *ontology-profiles* handled in AVATAR improve greatly the plain lists adopted in previous approaches. These lists allows to define the user preferences, but not organize them in a structural and hierarchical way, causing that discovery of new knowledge is clearly hampered.

On the other hand, to maintain the user preferences permanently updated, our proposal defines three indexes related to each class and each instance contained in his/her profile. These indexes are updated in an automatic way from the actions carried out by the TV viewer as we describe in [13]. They are the *DOI* (*Degree Of Interest*), the *Confidence* and the *Relevance* indexes. The former one measures the level of interest referred to each class/instance; the second one quantifies the success or failure of the system in the previous recommendations; and, finally, the *Relevance* index —which combines the previous ones— is used to rank the suggested programs. Finally, note that user profiles store both those contents the user likes (called *positive preferences*) and those he/she dislikes (*negative preferences*).

4 A Recommendation Strategy based on Semantic Inference

Once formalized the TV ontology and the user profiles, we describe the cornerstone of our recommendation process: the *semantic associations* supported in our approach.

4.1 The Semantic Associations

The *semantic associations* included in our approach are built taking as a starting point the relations established between the properties, and between the property sequences defined in $\mathcal{O}$. First, we formalize the information used to define these relations.

Let $PS_1 = [P_0, \ldots, P_m]$ and $PS_2 = [Q_0, \ldots, Q_m]$ be two property sequences, and ps_1 and ps_2 two instances of them, respectively.

Our approach defines a relation between any pair of properties as follows:

- ***Semantic Nexus between Properties*** ($\leftrightarrow_\rho$): P and Q are related by a *semantic nexus* —denoted as $P \leftrightarrow_\rho Q$— if either of the conditions is satisfied: (i) they the same property, (ii) P is superproperty of Q, (iii) Q is superproperty of P or (iv) P and Q are sibling properties.

Example 1: In Fig. 1 we can establish several semantic nexus between properties: for instance, *producerOf* $\leftrightarrow_\rho$ *presentedOf* is true because both properties are sibling. Analogously, *hasSupportingActress* $\leftrightarrow_\rho$ *hasStarringActor* is also verified.

Regarding the property sequences, two relations between them can be defined:

- ρ ***- Isomorphic Property Sequences*** ($\cong_\rho$). PS_1 and PS_2 are *$\rho-$isomorphic* —denoted as $PS_1 \cong_\rho PS_2$— if all their properties are related by *semantic nexus* one-to-one, that is: ∀ i, $0 \leq i \leq m$, $P_i \leftrightarrow_\rho Q_i$ is verified.

Example 2: Let us consider the sequences shown below. For clarity, the classes are represented in italics, and the properties in brackets.

PS_1 : *Soap Operas* [hasSupportingActress] *Actresses* [presenterOf] *Quiz Shows*

PS_2 : *Action Movies* [hasStarringActor] *Actors* [producerOf] *Terror Movies*

Taking into account the *semantic nexus* shown in Example 1, it is not hard to see that the above sequences are *ρ-isomorphic*.

- ***Joined Property Sequences*** ($\bowtie_\rho$). PS_1 and PS_2 are *joined* —$PS_1 \bowtie_\rho PS_2$— if they contain at least a common class. In other words, $PS_1 \bowtie_\rho PS_2$ is verified if PS_1.*NodesOfPS()* ∩ PS_2.*NodesOfPS()* is a nonempty set. This way, a join class C (i.e. $C \in PS_1$.*NodesOfPS()* ∩ PS_2.*NodesOfPS()*) is named *join node*.

Example 3: The sequence properties PS_1: *Terror Movies* [hasPlace] *Sport Places* and PS_2: *Sporting Events* [hasPlace] *Sport Places*, share the join node *Sport Places*, and consequently they are *ρ-joined*.

Taking into account the aforementioned relations, [12] defines significant *semantic associations* between two entities defined in a RDF(S) set. Some of them are especially useful in our recommendation process, and, for that reason, they have been assumed in our approach. A brief review of such associations is presented here:

I. ρ **-** ***pathAssociated***: ρ - *pathAssociated* (x,y) is true if there exists a property sequence instance ps and, either x and y are the origin and terminus of ps respectively, or vice versa, i.e. y is origin and x is terminus.

Example 4: We can illustrate this kind of semantic association by resorting to any of the property sequences shown in previous examples. So, we can define the sequence ps_1 by extracting specific instances of the classes shown in PS_1 in Example 2. This way, it is possible to define an association *ρ-pathAssociated* between the instance corresponding to the soap opera represented in Fig. 1, and the instance referred to the quiz show.

II. ρ **-** ***joinAssociated***: Let PS_1 and PS_2 be two joined property sequences with a join node C (i.e. $PS_1 \bowtie_\rho PS_2$). Besides, assume that there exist ps_1 and ps_2 which contain two instances —equal or different— belonging to this class C.

ρ - *joinAssociated* (x,y) is true if (i) x is the origin of ps_1 and y is the origin of ps_2, or (ii) x is the terminus of ps_1 and y is the terminus of ps_2.

Example 5: Considering the joined property sequences PS_1 and PS_2 defined in Example 3, it is possible to define two sequences ps_1 and ps_2 by extracting specific instances of the classes *Terror Movies*, *Sporting Events* and *Sport Places*. So, an association of the form *ρ-joinAssociated* is established between the terror movie represented in Fig. 1, and the sporting event. Note that the join node is the *Sport Places* class, however ps_1 y ps_2 contain different instances of this class: ps_1 is related to the stadium Amsterdam Arena and, on the other hand, ps_2 is referred to the stadium Yokohama.

III. *ρ* - ***isoAssociated***. *ρ - isoAssociated* (x,y) is true if there exist two *ρ-isomorphic* property sequences PS_1 and PS_2 ($PS_1 \cong_\rho PS_2$), and there exist ps_1 and ps_2 whose origins are *x* and *y*, respectively.

Example 6: This example is easy to extract from the *isomorphic* property sequences PS_1 and PS_2 shown in Example 2. So, an association of the form *ρ-isoAssociated* is established between specific instances corresponding to the soap opera starring Julia Roberts, and the action movie where Clint Eastwood is involved (in Fig. 1, *ρ - isoAssociated* (*Soap Operas*, *Action Movies*)).

With the goal of enhancing the inferential processes in AVATAR, several semantic associations not considered in [12], are defined in our proposal. Taking into account the necessary balance between the personalization and the diversification of recommendations, we propose the so-called *mixed semantic associations*, by taking as a starting point the *ρ-pathAssociated* association, given that it is the most direct and meaningful relationship out of the ones described in [12]. These *mixed associations* are shown next.

IV. *ρ* - ***join-pathAssociated***. Just as its name suggests, this association mixes the following ones: *ρ - pathAssociated* and *ρ - joinAssociated*. So, *ρ - join-pathAssociated* (x,z) is true if either of the conditions is satisfied:

(*ρ - pathAssociated* (*x*,*y*) and *ρ - joinAssociated* (*y*,*z*)) or
(*ρ - joinAssociated* (*x*,*y*) and *ρ - pathAssociated* (*y*,*z*))

Example 7: In Fig. 1, the associations *ρ-pathAssociated* (*Formula 1 Broadcasting*, *Sport Documentaries*) and *ρ-joinAssociated* (*Sport Documentaries*, *Soccer Broadcasting*) can be identified. Consequently, the association *ρ-join-pathAssociated* (*Formula 1 Broadcasting*, *Soccer Broadcasting*) is direct.

V. *ρ* - ***cp-pathAssociated***. Similarly, *ρ - cp-pathAssociated* (*x*,*z*) is true if:

(*ρ - pathAssociated* (*x*,*y*) and *ρ - cpAssociated* (*y*,*z*)) or
(*ρ - cpAssociated* (*x*,*y*) and *ρ - pathAssociated* (*y*,*z*))

VI. *ρ* - ***iso-pathAssociated***. Finally, *ρ - iso-pathAssociated* (*x*,*z*) is satisfied if:

(*ρ - pathAssociated* (*x*,*y*) and *ρ - isoAssociated* (*y*,*z*)) or
(*ρ - isoAssociated* (*x*,*y*) and *ρ - pathAssociated* (*y*,*z*))

4.2 The Process of Recommendation

The open and modular architecture proposed for AVATAR [14] allows to add new recommendation strategies without including significant modifications. For its novelty in the personalization domain, here we focus on the strategy based on semantic inference proposed in this paper. Regarding this, two phases can be distinguished:

- **Phase of Semantic Inference**: First, AVATAR explores the property sequences defined in $\mathcal{DB}_{\mathcal{O}}$ whose origin are instances contained in the user *ontology-profile*. The associations supported in our proposal are easy to infer by stemming from these sequences. For that purpose, it is only necessary to check if the conditions shown in Sect. 4.1 for each kind of association, are verified by these sequences. For example, two sequences which share a common class originate an association of the form ρ-*joinAssociated* between their respective origins (or terminus).

It is worth noting that in order to avoid retrieving a massive amount of meaningless associations, our approach resorts to a filtering methodology previous to the aforementioned inference. This methodology ignores those instances of the analyzed sequences which are not relevant enough according to the personalization requirements, and retrieves only significant associations. Obviously, the quantification of such relevance depends on the user preferences: if an instance is closely related to the programs the user liked, this one is not filtered and consequently, it will be contained in the inferred semantic associations. This phase returns a set of programs semantically related to the user preferences, together with the type of discovered association.

- **Phase of Ranking**: This list of programs must be ranked and shown to the users. In this process AVATAR considers two factors. The first one is the index *Relevance* of each instance in the user profile (remember Sect. 3.2). Note that if a program is very appealing for the user, its relevance index takes high values. This way, those inferred programs which are related to the user preferences with highest relevance, will be rank in provisional top positions of the suggestion. To confirm this position in the recommendation, AVATAR resorts to some parameters relevant in a personalized TV environment, such as the age rating of the program, the viewer favorite time for watching TV, etc.

5 Conclusions and Further Work

In this paper we have proposed AVATAR, a recommender system which takes the advantage of the Web Semantic and extends it to the domain of personalized TV. Its main contribution is a reasoning process about the user preferences and descriptions of TV contents. The recommendation strategy applied in our tool is based on suggesting a user TV programs which are semantically related to those watched in the past. Our approach differs from other existing proposals because semantic inference capabilities have never been considered in a personalization environment. Such inference process allows AVATAR to adjust in a flexible way to the kind of suggestion required by each viewer: if the user wants to watch programs similar to ones he liked in the past, AVATAR infers very direct associations between the suggested programs and his/her preferences. On the contrary, if the user does not mind watching varied programs, less similar to those he knows but always related to them, our inference methodology considers indirect associations. So, AVATAR discovers contents especially appealing for the user which would not be suggested by previous approaches.

It is worth noting that our inference methodology is not only valid for the TV domain. As mentioned in Sect. 4.2, the phase of inference reasons about a generic OWL ontology and the user preferences, by discovering semantic associations meaningful according to his personal interests. On the contrary, those parameters specific of the TV domain are considered in a separate phase (the phase of ranking). Such decomposition favors the reuse of our inference approach in other personalization applications.

For example, our proposal could be applied in recommender systems for the Semantic Web, where few proposals have been defined up to now. So, a possible example for motivating the usefulness of our approach in the Semantic Web, could be a system which suggests to a user travel destinations, by considering relations between countries according to their customs, culture, etc. For that purpose, it is only necessary to conceptualize this new domain by an OWL ontology, and to know the user preferences.

At this moment, we have a first prototype of AVATAR by which appealing recommendations have been obtained. Our goal is to extend these encouraging results to a real scenario with a greater number of users. For that reason, we are working on statistical studies by considering different population groups with diversified characteristics, such as genre, age, cultural level, etc. This way, we evaluate the usefulness of the semantic inference as novel recommendation strategy both in the TV domain and out of it.

References

1. Ardissono L., Kobsa A. and Maybury M. In: Personalized Digital TV: Targeting Programs to Individual Viewers. Kluwer Academic Publishers (2004)
2. Davies J., Fensel D. and van Harmelen F. In: Towards the Semantic Web: Ontology-driven Knowledge Management. John Wiley & Sons (2003)
3. Burke R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction **12** (2002) 331–370
4. Lang K.: Newsweeder: Learning to filter netnews. In: Proc. of 12th International Conference on Machine Learning. (1995)
5. Armstrong R., Freitag D., Joachims T. and Mitchell T.: WebWatcher: A Learning Apprentice for the World Wide Web. In: AAAI Spring Symposium on Information Gathering. (1995) 6–12
6. Terveen L., Hill W., Amento B., McDonald D. and Creter J.: PHOAKS: A system for sharing recommendations. Communications of the ACM **40** (1997) 59–62
7. Konstan J., Miller B., Maltz D., Herlocker J., Gordon L. and Riedl J.: GroupLens: Applying collaborative filtering to Usenet news. Communications of the ACM **40** (1997) 77–87
8. Balabanovic M. and Shoham Y.: FAB: Content-Based Collaborative Recommender. Communications of the ACM **40** (1997) 66–72
9. Smyth B. and Cotter P.: A personalized television listings service. Communications of the ACM **43** (2000) 107–111
10. Difino A., Negro B. and Chiarotto A.: A multi-agent system for a personalized electronic program guide. In: Proc. of the Workshop on Personalization in Future TV (TV-02). (2002)
11. Kurapati K., Gutta S., Schaffer D., Martino J. and Zimmerman J.: A multi-agent tv recommender. In: Proc. of the Workshop on Personalization in Future TV (TV-01). (2001)
12. Anyanwu K. and Sheth A.: ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In: Proc. of the 12th International World Wide Web Conference (WWW-03), ACM Press, 1-58113-680-3/03/0005 (2003) 690–699
13. Blanco Y., Pazos J., Gil A., Ramos M., López M. and Barragáns B.: AVATAR: Modeling Users by Dynamic Ontologies in a TV Recommender System based on Semantic Reasoning. In: Proc. of the 3rd European Conference on Interactive Television (EuroITV-05). (2005)
14. Blanco Y., Pazos J., Gil A., Ramos M., García J., Díaz R., Fernández A., Barragáns B. and López M.: AVATAR: Advanced Telematic Search of Audiovisual Contents by Semantic Reasoning. In: Proc. of the 4th Workshop on Personalization in Future TV (TV-04). (2004)