Ontology-Based Tourism for All Recommender and Information Retrieval System for Interactive Community Displays

K. Alonso, M. Zorrilla
Vicomtech-IK4
San Sebastián, Spain
{kalonso, mzorrilla}@vicomtech.org

R. Confalonieri, J. Vázquez-Salceda
Universitat Politècnica de Catalunya
Barcelona, Spain
{confalonieri, jvazquez}@lsi.upc.edu

H. Iñan, M. Palau
Tech Media Telecom Factory
Barcelona, Spain
{hasier.inan, manel.palau}@tmtfactory.com

J. Calle, E. Castro
Universidad Carlos III de Madrid
Leganés, Spain
{fcalle, ecastro}@inf.uc3m.es

***Abstract*— This paper presents a multi-stage ontology-based touristic recommender system which offers: personalized suggestions to citizens and tourists, including those with special needs; and information concerning the suggested locations. The system´s suggestions are based on user profiles which are continuously updated via feedback obtained from past interactions. Users' preferences are deducted by means of profiles and they are used to create and to send queries to heterogeneous information sources. The results are ranked and presented to the user along with related information.**

Recommender System; Ontology; Tourism; Information Retrieval

I. Introduction

Nowadays it is becoming common to search for domain-related content and activities through Interactive Community Displays (ICDs), which are multimedia information points offering interactive services on the public thoroughfare [1, 2, 3]. Some examples of ICDs can be found in Aberdeen and Bristol (UK), the i-kiosks and i+ respectively, providing information to people living in or visiting the city. More recently, many other initiatives have been deployed in commercial malls and other public spaces such as Punts BCN (a Barcelona city council initiative to offer information on public services). However, these services are often isolated or designed with predefined, static sources, not actually exploiting the benefits offered by the World Wide Web, and they usually do not distinguish users as individuals, providing similar information to users with different characteristics.

This issue makes it difficult to satisfy users who are typically looking for the most appropriate suggestions, according to their requirements and desires on that specific moment. Although the user could gradually adjust the stated suggestions, this is not the desired interaction as there can be too many possible choices, and preventing the user from having to wait too long is a main challenge to be considered. Therefore, what users really need are tools capable to adapt their behavior according to each situation and user, hence providing the best achievable choices with a highly personalized flavour.

One concrete way to achieve an effective personalization is by means of user profiles, storing user preferences and requirements to be taken into account when the content is selected. Although user profiles use to represent a common practice in personalization systems, they intrinsically bring several issues [4]. For instance, anonymous users do not have an initial profile and a model for predicting initial user preferences is required to avoid the cold-start problem. Moreover, user profiles' representation should be rich enough to represent preferences which depend on contextual information, and sufficiently compact to be able to be processed in a fast way. In fact, content should be filtered according to user preferences and requirements and user feedback (being implicit or explicit) should be used to keep the user profiles updated.

In this paper we present a multi-stage ontology based touristic recommender and information retrieval system for ICDs. The system is capable to offer personalized suggestions to citizens and tourist including those with special needs. The recommendation process is managed by using semantic representation, preference handling methods and ambient intelligence. The adaptation is concerned with several issues typically encountered in the representation of user profiles, such as cold-start problem, context-dependent preferences representation, content filtering and user feedback. The approach includes a Profile Manager for predicting user unknown features (or preferences), reducing the need of querying the user and expanding the adaptation possibilities; a Preference Reasoner for handling context-dependent preferences; a Content Manager for creating flexible queries for retrieving and integrating the content of heterogeneous information sources, and a Feedback Manager for interpreting user feedback to refine the user profile to progressively enhance the suggestions provided.

The rest of the paper is organized as follows. Section II briefly describes the related works. After presenting the system architecture (Section III), the next sections present the different modules that compose the system (Sections IV-V-VI-VII). Section VIII points out some results and concludes the paper.

II. RELATED WORKS

As pointed in the introduction, there are different technologies working together in the system. Two main technologies can be pointed, recommendation technologies and information integration and retrieval technologies.

Recommender systems have been a very active research and application area. There are various application examples that suggest movies [5], news [6, 7], music and books [8].

The recommendation techniques can be divided into four different groups [9]:

- **Collaborative:** These recommender systems only use for their recommendations the rating profiles from different users [6, 10].
- **Content-based:** The recommendations are generated by the features associated with the products and the users´ ratings [11, 12].
- **Demographic:** The results of this kind of systems are based in demographic profiles of the users. They use the users´ ratings from specific demographic niches [13].
- **Knowledge-based:** A knowledge-based systems infers their recommendations based on users´ needs and preferences [14, 15].

Hybrid recommender systems are those that combine two or more of the techniques listed above to improve recommendation performance [7, 9, 16, 17, 18].

Nowadays a great number of search tasks in complex information systems require the participation of multiple information sources. The information is usually scattered over the web in different sources implemented in a different technology, and with a different structure [19]. In the last years, there have been many contributions that employ ontology-based semantic approaches to improve the access and the integration of heterogeneous information sources. The Semantic Web [20] deployment is a slow but constant process, and there are powerful technologies in niche applications (healthcare, finance, publishing among others).

In the touristic domain, the scientific community has provided many relevant works which use ontologies to retrieve information. The REACH project implements an ontology-based representation to provide enhanced access to heterogeneous distributed cultural heritage information sources [21]. Another good example can be found in E-Tourism project [22] that develops an ontology-based system to improve information creation, maintenance and delivery in the touristic industry by introducing semantic technologies.

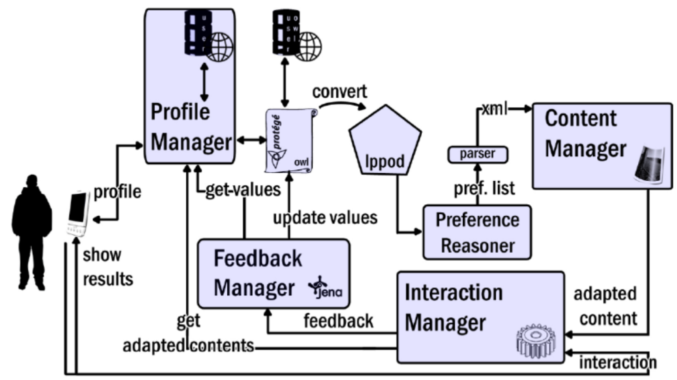


Figure 1. System architecture

III. SYSTEM ARCHITECTURE

In the Fig. 1 the recommender system architecture is shown. It is composed by different modules that provide capabilities to automatically adapt user profiles and retrieve the most accurate information according to the recommendations extracted from the user's profile and context information, both of them defined in the general ontology whose structure is defined in Fig. 2, which shows the restaurants´ use case.

The Profile Manager deals with the representation of user preferences. At the beginning an initial user profile is created from a user group. The profile is stored according to the profile ontology which contains weighted contextual preference relations. New user profiles are created inheriting preferences from predefined user groups. As next, the abstract user profile representation is adapted to the requirements of the Preference Reasoner. This module collects the current context and reasons about the context-aware preference rules. The reasoning generates an ordered list of context-aware preference items. Thereafter, such preferences are processed by the Content Manager which is in charge of filtering the appropriate content to be provided. Finally, once the suggestions are provided, the user can express some feedback that is processed by the Feedback Manager to update the user's profile.

IV. PROFILE MANAGER

The Profile Manager represents the users´ profiles and is supported by a User Model [23], that is, a knowledge base and a set of reasoning mechanisms regarding user characterization in order to outline their preferences and requirements and adapt the interaction to them. The knowledge base contains facts relating a user (or a group of users [24]) with a user feature and a value (within that feature domain). A user feature might be any perceivable attribute or behavior of the user, observed within the user profile ontology described in the previous section. The main reasoning mechanisms are usually the following two: i) fit, i.e., decide in which group within the base the current interlocutor should be included; and ii) inference, i.e., predict the value a feature takes for the current user. Group User Models based on experience frequently observe a third process, the fusion, aimed to include a new user into a group, or to merge two groups.

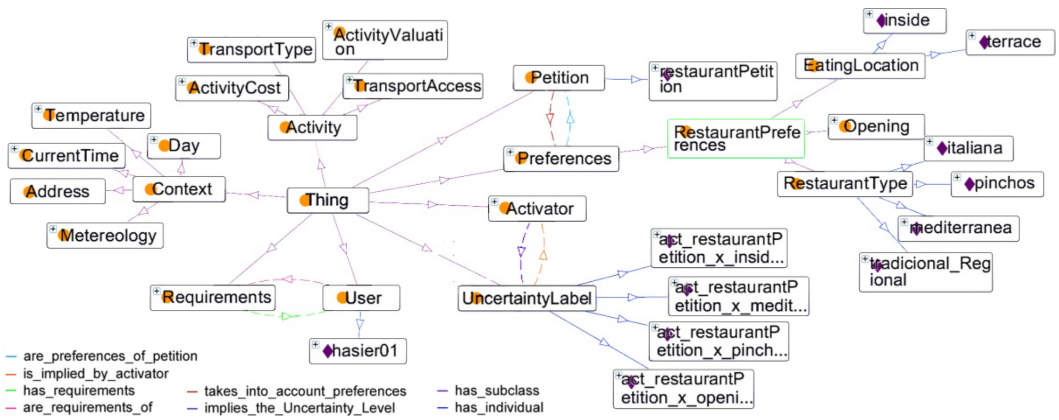


Figure 2. Ontology overview including some instances for the restaurants´ use case

Specifically, this system counts on a predictive statistical user model [25]. Its knowledge base contains descriptions of user groups within the domain. As new information about user characterization is acquired from the interlocutor, the user will be matched within a group. Anytime the Profile Manager requires a value for some unknown feature, the User Model can provide it (along with a certainty value regarding the prediction accuracy). Adaptation rules are based upon those features, so interaction can be adapted to the interlocutor even if the user is not well known. Initially, when the interlocutor is completely unknown, any predicted value is the most plausible for that feature in the most probable group. The certainty regarding those predictions will be probably too low at that time, so any touchy adaptation rule based on them will be discarded.

However, rules regarding more general behavior (where mistakes are less important) can be triggered with less certainty, thus avoiding the cold-start problem. As the interaction is performed, facts on interlocutors' features and behavior are acquired, and consequently new predictions will be directed by that information.

The Profile Manager observes identifiers, but the User Model does not. Provided that there is no user identification throughout the knowledge base, there cannot be records of specific interlocutors but general stereotypes. In other words, any new session involves a new user, no matter how many times he has previously interacted with the system (yet the system will have recorded all those sessions as other similar users, and will take them into account for this new session). Consequently, dynamic features will be observed as completely volatile, and so will be the adaptive behavior based upon them. For example, the same user with different moods can be seen as different users, and some adaptation may vary.

This User Model is populated through the experience, that is, each session held with a user augments the knowledge base in order to improve future uses. The information acquired from the interlocutor is anonymized and shaped into a stereotype which will be stored as a new group or used to refine an existent one. However, the model supports an initial knowledge load. In those cases where experts are available to analyze the population of potential users in the interaction domain, a set of default user groups (properly weighted) can be introduced in the knowledge base. Using the system, as experience is gained, right groups will be strengthened while the less successful ones will be weakened or may even eventually disappear (a weight factor determines the pace for this process).

The Profile Manager can rely on the User Model in those cases where there is a lack of required information and interrupting the conversation to get it is not adequate (it involves the risk of annoying the user). If the provided value is good enough, it can be applied with less cost (the mistaking risk). Feedback (either explicit or implicit) plays a key role in refining the interlocutors' profile, and ultimately increases the quality of knowledge stored in the database. For example, some past users who moved their hands slowly across the touch screen of the ICD requested bigger images (zoomed in) when the system showed them maps. Now the system has to display a map, and since it should not ask details such as the map size or zoom, it will take a decision based upon the user models advice. If the current interlocutor moves his hand slowly across the screen, the recommendation will probably be to zoom in, due to the available facts.

V. PREFERENCE REASONER

The Preference Reasoner is a logic programming module that takes user's profiles as input and generates an ordered list of context-aware preferences. In order to process the user profile in a logical way, the profile is first converted from its ontological representation into the appropriate syntax accepted by the reasoner. At the symbolic level, a user profile is represented according to an extended syntax of Logic Programs with Possibilistic Ordered Disjunction (LPPODs) which is a recently defined framework able to represent and reason about context-dependent weighted preferences [26].

Formally, an LPPOD P is a logic program composed of a finite set of possibilistic ordered disjunction rules (preference rules for short) of the form $(w_1 : p_1) \times\ldots\times(w_k : p_k) \leftarrow c_1 \wedge \ldots \wedge c_m \wedge not\ c_{m+1} \wedge \ldots c_m \wedge not\ c_{m+n}$, in which the p_i's are preference literals, c_j's are literals are context literals and w_i's are weights belonging to a finite linearly ordered scale [0, 1, … , 100]. Each w_i measures the importance of preference p_i in a preference rule. The intuitive reading of a preference rule is *if possible, p_1 is preferred with weight w_1*, or *if p_2 is not possible, then p_2 is preferred with weight w_2*, and so on. By convention, the left and right parts of the rule are known as head and body respectively. A preference rule is satisfied when its body is satisfied, i.e., the $c_1, \ldots, c_m$ are true and there is no evidence about $c_{m+1}, \ldots, c_{m+n}$.

In our approach, preference rules are used to represent context-dependent preferences and a program P for representing a user profile. We map p_i's to user preferences and c_j's to contextual information. In such a way, we are able to represent that, in a given context, a user has an ordered set preferences associated with different weights. The higher the w_j, the higher the importance of a preference p_i (for instance a requirement is associated with the maximum value, i.e., 100).

The satisfaction of each rule basically depends on the presence of contextual information, i.e., only the preferences of satisfied rules are taken into account in the reasoning process. As such, only those preferences, the user is contextually interested in, will appear as program solutions (in accordance to the LPPOD semantics definition [26]).

Each solution of an LPPOD program is a list of $\langle p,w \rangle$ and $\langle c \rangle$ where p is a preference, w is a weight, and c is a context. Usually, an LPPOD has several solutions, and each of them consists of a list of context-dependent preferences.

One of the distinctive characteristics of LPPODs is the possibility to specify an order among its solutions. The order among preferences depends on the position of the best satisfied preference literals and it is defined by a Pareto-based comparison criterion [26].

To exemplify the use of LPPODs, let us consider a (simple) user profile expressing the preferences of a user. The user prefers *to eat in a Mediterranean restaurant rather than in a Vegetarian and if it is between 15h and 18h he/she wants to take a coffee*. The profile can be encoded by the program $P = \{r_1 = (90 : type\_Med) \times (50 : type\_Veg) \leftarrow pet\_Rest, r_2 = (80 : type\_CoffeShop) \leftarrow pet\_Rest: time\_15 - 18h\}$. In context (pet_Rest; 100), r_1 is satisfied, while r_2 is not. Therefore, the solutions associated with this profile are $S_1 = \{(type\ Med: 90), (pet\ Rest: 100)\}$ and $S_2 = \{(type\_Veg: 50), (pet\_Rest: 100)\}$. By considering the preferences positions in the rule, it can be easily checked that solution S_1 is preferred to solution S_2.

The solutions of an LPPOD are computed by means of an Answer Set Programming-based solver [27]. The output of the Preference Reasoner is a sorted list of preference items..

VI. CONTENT MANAGER

The Content Manager's process starts with the preference reasoner output. This output combines user preferences with a different number of context-aware preference elements (Fig. 3(a)), which are the key to retrieve valuable information from data storage units. The content-aware preferences are related with the stored content, but not directly. For this reason, before querying the data storage units, it is mandatory to proceed with a parsing process. This allows the correct matching between the query and the stored information.

The Content Manager integrates an ontology which drives the parsing process. The Content Manager's ontology defines the parsing options for every preference, linking context-aware preference individuals with the related individual concepts in the data storage units, and guides the querying process, signaling the correct repository to send the query and what type of query must be constructed (SPARQL in semantic repositories, SQL in data bases, etc) besides the skeleton of it. For instance, in Fig. 3(a), among other preferences, a user with deafness as functional diversity requests for information on restaurants. The ontology relates these preferences with parameters as sign language staff and round dining tables. Thus, in the querying process, the Content Manager will request restaurants that have at least one staff member with sign language knowledge, and restaurants with round dining tables in their dining room.

Content Manager Input											
Lunch		*Dinner*		*Transport Type*		*Transport Access*		*Activity Cost*		*Activity Valoration*	
Value	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*
13-14	80	20-21	70	Metro	75	L3	75	1	75	5	90

Content Manager Input									
Restaurant Type		*Opening*		*Eating Location*		*Functional Diversity*		*Gastronomic Requie.*	
Value	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*
Traditional	80	7	70	Terrace	90	Deaf	100	Vegetarian	80

(a) Example of context-aware preference elements

Response	*Restaurant Type*		*Terrace*		*Functional Diversity: Deaf*				*Vegetarian*		*Total*
					Sign Language		*Round Table*				
	Value	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*	*Value*	*Weight*	
Sin Pa	Traditional	80	True	90	True	20	True	80	True	80	350
El Toison Dorado	Traditional	80	True	90	False	0	True	80	True	80	330
La Bocateria	Traditional	80	True	90	True	20	True	80	False	0	270
Bola 8	Traditional	80	True	90	False	0	True	80	False	0	250
XXL	Traditional	80	False	0	False	0	True	80	False	0	160

(b) Example of personalized content

Figure 3. Context-Aware Preferences and Personalized Content

The Content Manager sends different queries taking into account each preference. The responses of these queries are analyzed and internally stored in a list of results. Two scenarios are possible: (i) the obtained result instance is not in the result list. In this case, the result is added to the list with its corresponding ranking value of the query type; (ii) the obtained result instance is already in the result list.In this case the ranking value of the result is updated adding the corresponding ranking value of the query type.

The ranking value for each query type depends on the type of parsing. Two different parsing types exist taking into account the relations between the preference concept and stored information concept. The procedures to assign ranking values are also different:

- **one-to-one relationship:** a preference concept match directly with a stored information concept. The ranking value is equal to the preference weight.
- **one-to-many relationship:** a preference concept involves two or more stored information concepts. The ranking value is based on the preference weight and then it is weighted again in a percentage depending on the relevance of the stored information concept. This percentage is defined in the Content Manager's ontology as stored information concept

data property. The value of this percentage is fixed combining the results of a survey with future end users and the analysis of the results obtained with different percentage in several queries. For the example shown in Fig. 3(b), the stored information concepts related with deafness are weighted as Staff member with sign language knowledge, 20% and Round dinner tables in the dining room, 80%.

Finally when all queries are processed, the results are ranked in descending order using the ranking value as reference. Fig. 3(b) resumes the partial ranking values obtained by a list of restaurants for some preferences and the output order obtained by their sum.

VII. Feedback Manager

The Feedback Manager is responsible for adapting user profiles by processing interactions of the users and valuations on provided suggestions. The evaluation process follows two modes: (i) explicit evaluation, contemplating an explicit rating of the user on each suggestion; (ii) implicit evaluation, considering user selections on the provided suggestions as positive feedback. In this case, preferences related with such suggestions are rated with a default value.

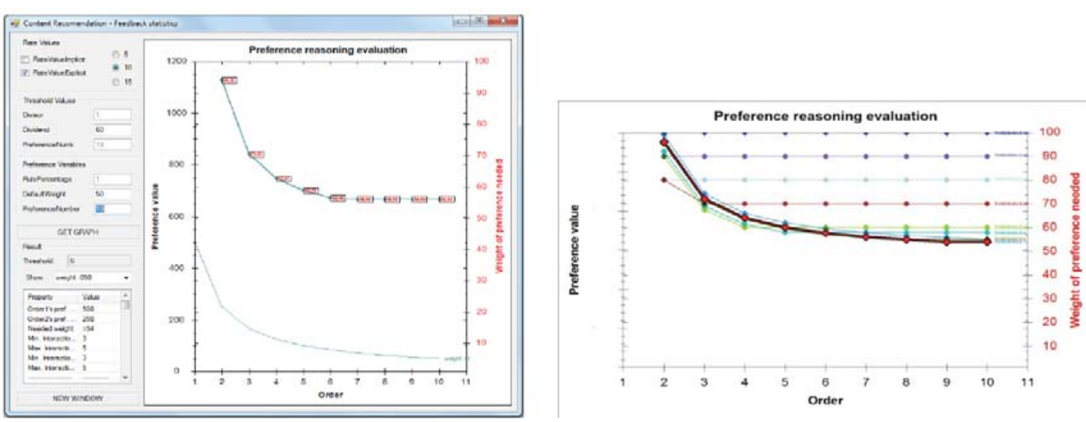
Once the evaluation is done, the Feedback Manager receives a list of distinct rankings of each preference evaluated in a set of contexts. Thus, the module interacts with the Profile Manager to update or create new preference rules based on a given context set. At this point two scenarios are possible: (i) if the user profile already contains the preferences for the current context, these are updated increasing or decreasing their weights. This may imply changes on the position among preferences during the preference reasoning; otherwise, if such preference does not exist for the current context, a new preference rule with such context is added to the user profile.

Context-aware preferences are prioritized depending on their weights and positions inside the preference rule. Thereby, depending on the rate given in the feedback process, the preference weight is increased or decreased by some static values. Once the preferences weights are updated, a threshold is applied to change the preference position. This arrangement is performed according to the relation between weight and position compared with their successor or ancestor values, with a margin of previously computed threshold values ($\delta = \frac{dividend}{divisor * pref\_num}$).

While divisor and dividend are static values, pref_num is the number of preferences that take part in the rule. In order to trigger a change in the position, the next comparison is done: if $|\ (pv_2 - pv_1)\ |\ pv_1\ | \leqslant \frac{pv_1 * \delta}{100}$) then *changePosition*($p_1$: p_2). The preference value p_i (pv_i) is a relation of weights and its position in comparison with the number of preferences in the rule ($\mathrm{pv_i} = \frac{\mathrm{w_i}}{\mathrm{i * pref\_num}}$).

An evaluation of possible values has been performed, graphically representing required weight on preferences to increase their position in a rule. The number of preferences and minimum interactions needed to perform the aforesaid position to change are also taken into account, as seen in Fig. 4(a). After evaluating several results, the following conclusions have been drawn:

- The lower the threshold value the bigger the weight that the preference needs to overcome its ancestor, due to the fact that the preference value will be based almost on their positions. In cases where the threshold is too low, preferences positioned first will need weights close to 100 (maximum value) to be reordered; by contrast, preferences placed in last positions will need almost the same weight as the ancestor preference (minimum value).
- Focused in the results on the graphic shown in Fig. 4(b), an exponential trend is considered as optimum, where a reasonable interaction number is taken also into account when rating updated weight values. The explanation of this trend is as follows: preferences located in the rules' tail have less difficulties to be reordered (as lower weights are needed, changes occurred more often) as they are considered less important; in the same way, first placed preferences have more difficulties to change their position in the rule, as they are considered essential in that sub-domain because bigger weights are needed (changes happens less often).



(a) Distinct threshold on same preference weight comparison (b) Number of preference and threshold relation

Figure 4. Relationships between preference weights and threshold values

VIII. Results, Conclusions and Future Research

In this paper we have presented a multi-stage ontology based touristic recommender and information retrieval system for ICDs. The system is capable to offer personalized suggestions to citizens and tourism including those with special needs. The adaptation of user profiles and the recommendation of content and activities are based on the user interactions according to several factors such as profile assignment, content filtering and user feedback. In cases where anonymous users use the service, Group User Models prevent the cold-start problem. However, in any case, the user profile must be continually adapted based on user interactions and feedback, due to the fact that the initially assigned profile is not close enough to provide user personalized content. Anyhow, the ICD service manages the user profile creation and the maintenance of preferences so that the user can change the assigned preferences and requirements.

As discussed in the previous section, an exponential tendency in required preference weights characterizes the change among preferences orders, depending on updated preferences position: less thrust on rules head (as they are considered more important) and more thrust on rules tail (as they are less important or they are new preferences to be considered in that context or petition). Since new context-aware rules can be created and updated in each feedback process, the more feedback received, the more adapted the profile is concerning to the user necessities; furthermore, fewer changes will be performed in future feedback since all given preferences will be in rules first positions. Thereby, the system provides the explicit feedback feature which could mean a personalized rating, and therefore, more valuable feedback (higher weight updates) than systems implicit feedback.

The contents offered are based on the selection of preferences obtained from the Preference Reasoner which makes the computation time proportional to the number of rules and preferences found in the user profile (since all those rules and sentences are processed). In the same way, the Content Manager processes, based on information sources, are performed almost immediately even though it also depends on the number of preferences received. The whole system usability and accessibility evaluation will be carried out later on. It will help to improve ICD interface, used preferences and the weights given to each queries in order to upgrade the quality of the results.

Acknowlegment

This work has been partially supported by the SEMANTS project (TSI-020110-2009-419) which is partially funded by the Spanish Ministry of Industry, Tourism and Trade.

References

[1] L. Ceccaroni, V. Codina, M. Palau y M. Pous, «PaTac: Urban, Ubiquitous, Personalized Services for Citizens and Tourists.,» 2009.

[2] I. Gómez-Sebastià, M. Palau, J. C. Nieves, J. Vázquez-Salceda y L. Ceccaroni, «Dynamic Orchestration of Distributed Services on Interactive Community Displays: The ALIVE Approach.,» 2009.

[3] M. Palau, L. Ceccaroni, I. Gómez-Sebastiá, J. Vázquez-Salceda y J. C. Nieves, «Coordination and Organisational Mechanisms Applied to the Development of a Dynamic, Context-aware Information Service,» 2010.

[4] G. Adomavicius, E. Tuzhilin y E. Tuzhilin, «Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,» 2005.

[5] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, J. Riedl y J. Riedl, «MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System,» 2003.

[6] A. Das, M. Datar, A. Garg, S. Rajaram y S. Rajaram, «Google News Personalization: Scalable Online Collaborative Filtering,» 2007.

[7] L. Li, D. Wang, T. Li, D. Knox, B. Padmanabhan y B. Padmanabhan, «SCENE: a scalable two-stage personalized news recommendation system.,» 2011.

[8] G. Linden, B. Smith, J. York y J. York, «Amazon.com Recommendations: Item-to-Item Collaborative Filtering.,» 2003.

[9] R. Burke, «Hybrid Recommender Systems: Survey and Experiments,» *User Modeling and User-Adapted Interaction,* vol. 12, nº 4, pp. 331-370, 2002.

[10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl y J. Riedl, «GroupLens: An Open Architecture for Collaborative Filtering of Netnews,» 1994.

[11] E. Gabrilovich, S. Dumais, E. Horvitz y E. Horvitz, «Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty,» 2004.

[12] W. IJntema, F. Goossen, F. Frasincar, F. Hogenboom y F. Hogenboom, «Ontology-based news recommendation.,» 2010.

[13] B. Krulwich, «Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data,» 1997.

[14] M. L. Jensen, P. B. Lowry, J. K. Burgoon, J. F. Nunamaker y J. F. Nunamaker, «Technology Dominance in Complex Decision Making: The Case of Aided Credibility Assessment.,» 2010.

[15] R. B. Recommender, R. Burke y R. Burke, «The Wasabi Personal Shopper: A Case-Based Recommender System,» 1999.

[16] J. Liu, P. Dolan y E. R. Pedersen, «Personalized news recommendation based on click behavior,» 2010.

[17] S.-S. Weng, B. Lin, W.-T. Chen y W.-T. Chen, «Using contextual information and multidimensional approach for recommendation.,» 2009.

[18] M. J. Pazzani, «A Framework for Collaborative, Content-Based and Demographic Filtering,» *Artif. Intell. Rev.,* vol. 13, nº 5-6, pp. 393-408, 1999.

[19] Y. Arens, C. Y. Chee, C.-N. Hsu y C. A. Knoblock, «Retrieving And Integrating Data From Multiple Information Sources,» *International Journal of Intelligent and Cooperative Information Systems,* vol. 2, pp. 127-158, 1993.

[20] T. Berners-Lee, J. Hendler y O. Lassila, «The Semantic Web,» *Scientific American,* vol. 284, nº 5, pp. 34-43, 2001.

[21] C. Doulaverakis, Y. Kompatsiaris y M. Strintzis, «Ontology-Based Access to Multimedia Cultural Heritage Collections - The REACH Project,» 2005.

[22] C. K. G. y M. SaravanaPriya, «Ontology Based Information Retrieval for E-Tourism,» *International Journal of Computer Science and Information Security,* vol. 8, pp. 78-83, 2010.

[23] A. Kobsa, «Generic User Modeling Systems,» *User Modeling and User-Adapted Interaction,* vol. 11, nº 1-2, pp. 49-63, 2001.

[24] T. Finin y D. Drager, «GUMS: A General User Modeling System,» 1986.

[25] I. Zukerman y D. W. Albrecht, «Predictive Statistical Models for User Modeling,» *User Modeling and UserAdapted Interaction,* vol. 11, nº 1, pp. 5-18, 2001.

[26] R. Confalonieri, J. C. Nieves, M. Osorio, J. Vázquez-Salceda y J. Vázquez-Salceda, «Possibilistic Semantics for Logic Programs with Ordered Disjunction.,» 2010.

[27] N. J. Vázquez-Salceda, «Towards the Implementation of a Preference- and Uncertain-Aware Uncertain-Aware Solver Using Answer Set Programming.,» 2010.