

Ontology-based User Preferences and Social Search for Spoken Dialogue Systems

Yves Vanrompay, Nesrine Ben Mustapha and Marie-Aude Aufaure
MAS Laboratory, Ecole Centrale Paris
Grande Voie des Vignes, F-92 295 Chatenay-Malabry, France
Email: {Yves.Vanrompay, Nesrine.BenMustapha, Marie-Aude.Aufaure}@ecp.fr

Abstract—Many current spoken dialogue systems for search are domain-specific and do not take into account the preferences and interests of the user. In order to provide a more personalized answer tailored to the user needs, we propose a spoken dialogue system where user interests are expressed as scores in modular ontologies. This also allows us to cover multiple domains (e.g. searching for restaurant, housing, ...) because each ontology module corresponds to a search domain. This approach allows for a dynamic and evolving representation of user interests. Moreover, a collaborative search of users with similar interests allows to build ad-hoc communities where information can be shared amongst and recommended to users. We propose to use techniques borrowed from formal concept analysis to flexibly and efficiently build these communities.

I. INTRODUCTION

Our goal is to design and build mobile applications that approach human performance in conversational interaction, specifically in terms of the interactional skills needed to do so. These skills include recognising and generating conversational speech incrementally in real-time, adapting to new concepts without manual intervention, and personalising interaction. All of these skills will be learned or adapted using real data, and will be used to build systems for interactive hyper-local search for domains such as property search and tourist information. Current search engines work well only if the user has a single search goal and does not have multiple trade-offs to explore. For example, standard search works well if you want to know the phone number of a specific business but poorly if you are looking for a house with several different search criteria of varying importance, e.g. number of bedrooms versus bathrooms versus price etc. The latter requires the user to collaborate conversationally over several turns. Moreover, current spoken dialogue systems are mostly domain-specific, using rather static information from experts and knowledge bases.

We incorporate modular ontologies, where each ontology module represents a domain and can be dynamically loaded at runtime to meet the current needs of the user. In order to provide a personalized answer, tailored to the specific user, the concepts and attributes in these ontology modules are annotated with scores representing the preferences and interests of the user. This allows us to learn the specificities of a user, and give responses that fit the user's profile. Also, this provides us with the building blocks for constructing ad-hoc communities of similar users where information can be shared

and recommendations can be made. The use of collaborative filtering makes it possible to provide answers to the user queries which are even more tailored to his profile and which exploit information from similar users in the neighborhood.

This paper is organized as follows. Section 2 motivates our research by presenting an example dialogue between the system and the user. Also we give an overview of the different elements in our system. Section 3 introduces the representation of user interests as scores associated to the different elements in modular ontologies. Then, section 4 describes formal concept analysis techniques to dynamically derive suitable communities of interest. Section 5 presents related work, after which we conclude and give pointers to future research directions.

II. USER PREFERENCES REPRESENTATION AND SOCIAL SEARCH FOR SPOKEN DIALOGUE SYSTEMS

This section starts with an example dialogue between the user and the system motivating the need for personalized answering. Then, we give an overview of the different components of the proposal.

A. Motivating Scenario

This section illustrates the envisioned spoken dialogue system by a motivating scenario. Let us suppose that Paul is visiting Paris for the moment, and he is looking for a good restaurant in the center of the city. An example dialogue between Paul (U) and the system (S) can be as follows:

- 1) U: Hi, I'm looking for a descent restaurant not too far from where I am now.
- 2) S: Uhhm, let me see.... . There are some Indian and Chinese restaurants closeby.
- 3) U: Ok, what about the Indian restaurants?
- 4) S: Well, the closest one is on Boulevard St. Germain. It also has free Wifi.
- 5) U: Oh, I don't really like that street, too much traffic and noise.
- 6) S: I have another Indian restaurant called Tandoori, it is on Boulevard St. Michel. Price range is medium.
- 7) U: Ok, great, I'll take that one.

In this conversation, not only information on user interests learned from past dialogues is exploited, but also new information on interests and preferences is learned:

- (2) The system has learned from previous conversations that the user is quite fond of Indian and Chinese restaurants. Therefore, it starts with proposing restaurants with these food types.
- (3) The user's interest scores are updated in favour of Indian restaurants.
- (4) The system knows that the availability of Wifi is important to the user, so it proactively gives this information.
- (5) User preferences are updated: the user does not like Boulevard St. Germain.
- (7) The Indian restaurant Tandoori is ranked higher in the user preferences.

From the above listing we can see that the preferences of a user are expressed on different levels. In (2) and (4), information is given by the system on specific attribute types that are important to the user. (3) and (5) show that preferences are associated with particular attribute values. Finally, in (7), an interest in a concrete restaurant instance is expressed.

B. Proposal Overview

The starting point for inferring user interests are abstract representations of the history of dialogues (dialogue act units) between the user and the system. By analyzing this history, we derive interest scores associated to concepts, attribute types and attribute values in an ontology module corresponding to a specific domain (section 3). The weighted ontologies are the building blocks for the second part of our proposal, which is the social search (section 4). By combining the interests from several users and organizing this into a concept lattice, we can automatically identify communities of users which share similar interests. By using these communities, recommendations are made to users regarding their interests, i.e. users are recommended other communities which could be of interest.

III. USER PREFERENCES AS WEIGHTED MODULAR ONTOLOGIES

A. From Dialogues to User Preferences

User preferences and interests regarding concepts and attributes are inferred from the user's dialogue history. All user and system utterances from past dialogues are saved in a so called dialogue act unit (DAU) format, which is an abstract representation of the utterances. For example, when the user asks for the price range of a restaurant, this is represented as the DAU *request(price)*. As another example, when the user expresses that he wants Italian food, the corresponding DAU is *inform(food=Italian)*. DAUs can also express negative interests, like in *request(food != Indian)*. Interest scores are derived from logged traces of DAUs. We keep track of the positive and negative occurrences of attribute types, attribute values and concrete instances. These frequencies allow us to rank the different elements. Based on this ranking it is decided which system response is best suited with regards to the user interests. If the user often queries for pricing information in searching for a restaurant, the attribute type *price* will have a high frequency, and the system will proactively inform the

user about the price in its answers, without the user having to ask for it. If consequently the user often favours expensive restaurants, the attribute value *expensive* for the type *price* has a relatively high score, leading the system to recommend restaurants from a higher price range.

B. Weighted Modular Ontologies

Our mechanism to expressing user interests is integrated in an approach which represents information as (hierarchical) modular ontologies. Each ontology module corresponds to a particular domain. As can be seen in Figure 1, a tourism ontology can contain several ontology modules like lodging, transportation and restaurant information. Taking the restaurant ontology module, this contains (amongst others) the restaurant concept with attribute types name, food type, dress code and location. To personalize the responses given to the user, our user model incorporates the interests and preferences of the user by assigning scores to elements in the appropriate ontology modules. These scores are updated according to what is being learned from the history of past dialogues, ensuring that the interests evolve as user preferences may change through time. More concretely, ontology modules are associated with weight vectors expressing the interest or disinterest of a user on 4 different levels, which we explain example-wise by referring back to the motivating scenario.

- Concepts: At the highest level, the general interest of a user in a concept is expressed. For example, in the tourist modular ontology, Paul is interested in information on restaurants, but not in information on libraries.
- Attribute types: Here, interests on types of information associated with a concept are represented. For example, Paul typically queries on the food type of the restaurant, but not on the dress code or price level.
- Attribute values: Interests in particular values of an attribute are expressed. For example, concerning the food type, Paul specifically likes Chinese and Indian food.
- Instances: When the user has chosen a specific restaurant in the past, this is explicitly represented by a score associated with the instance representing that restaurant. Otherwise it is not possible to make a preference distinction between restaurants with identical attribute values, but which were effectively chosen or not by the user in the past. By explicitly choosing for the Tandoori restaurant, Paul's score for that restaurant is positively updated.

Weights in the four mentioned levels are aggregated to form scores signifying a user's (dis)interest in a specific instance. The weights are useful in two different aspects:

- The scores are used to rank and recommend concrete instances that are of interest to users.
- Attribute types and values are used to generate a system response tailored to the user needs. For example, based on the interest scores, the system can decide to proactively inform the user on the food type of the restaurant, but not on the dress code.

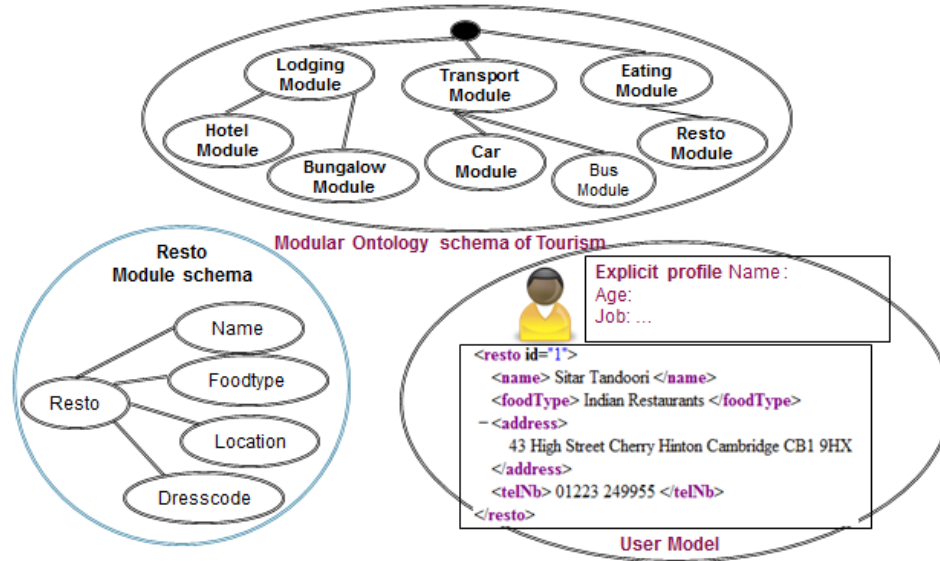


Fig. 1. Weights associated to the different elements in a restaurant ontology

IV. SOCIAL SEARCH WITH CONCEPT LATTICES

Although communities of users can be defined statically, this is not optimal because it restricts the kind of communities that can be formed. In general, it is not possible to know at design time what communities will be relevant in a given context. In order to provide a personalized experience to the user, in which communities can be dynamically recommended in function of the interests of the user, we propose the use of concept lattices. Interests that (potentially can) specify communities are represented as concepts in a Galois lattice, which allows conceptual clustering and the discovery of novel communities that match users needs. Lattice-based formal concept analysis takes as input a matrix specifying a set of users in the geographical neighborhood (the objects) and the corresponding interests (the properties or attributes) specifying the community each user belongs to. This representation allows us to:

- find natural user clusters: the set of all users that share a common subset of interests.
- find natural interest clusters: the set of all interests shared by one of the natural user clusters.

Natural interest clusters correspond with natural user clusters, and a concept can be seen as a pair containing both a natural property cluster (interests) and its corresponding natural object cluster (users).

By analyzing this emerging and evolving concept lattice, we gain the following features. 1) community membership and dependencies between communities can be inferred to dynamically form new communities suitable to the overall user interests. 2) we can efficiently recommend possibly interesting communities to users. The detection of potentially suitable communities uses similarity metrics between the interests specifying a community and the users interests, which are represented as scores associated with concepts and attributes in an ontology. Similarity between different users can be

calculated in order to recommend communities based on collaborative filtering.

A. Concept Lattices

We propose techniques based on formal concept analysis (FCA) to identify a) sets of users that share common interests and b) relations and dependencies between interests. Analyzing this information allows us to automatically derive appropriate communities.

In FCA, a concept is identified both by an extent, i.e. a set of objects that share the concepts attributes, and an intent, the set of attributes common to the concept. The starting point of FCA is a formal context, which is defined as a triple (G, M, I) , where G is a set of objects, M a set of attributes and I a binary relation over $G \times M$. A formal context corresponds to a binary matrix with the rows being the objects and the columns the attributes. Given $O \subseteq G$, O is defined as the set of attributes of M which are shared by all elements of O . Given $A \subseteq M$, A' is defined as the set of objects that share all attributes in A . A formal concept of a context is a pair (O, A) , O being the objects that share all attributes of A . O is the extent and A the intent, i.e. $O = A'$ and $A = O'$. The set of all concepts derived from the formal context can be represented as a lattice, based on the subconcept relation \leq for partial ordering: $(O_1, A_1) \leq (O_2, A_2)$ if $O_1 \subseteq O_2$ or equivalently that $A_2 \subseteq A_1$. The maximum element in this ordering is the concept with an empty extent, and the minimum is the concept with an empty intent.

B. Inference of Communities

As was explained in the previous section, the users interests and preferences are expressed in ontology modules as entities and attributes that are annotated with scores $-1 \leq w_i \leq 1$ indicating a users (dis-)interest in a particular item. Each ontology module is domain-specific, representing for example

interests in houses for sale or restaurants. For example, the restaurant sub-module contains attributes like location, food type and dress code. Scores associated with these attributes indicate whether the user prefers e.g. Indian or Italian restaurants, or whether the dress code should be formal or casual.

Each scored ontology module serves as input for the construction of the formal context (G, M, I). The set G holds all users, while M contains relevant attributes (interests). I is defined as follows:

$$I(g_i, m_j) : G \times M \rightarrow \{0, 1\} = 1 \text{ if } w_j(u_i) \geq T, \text{ otherwise } 0$$

$w_j(u_i)$ is the score assigned for user i to attribute j , and T is a threshold between 0 and 1. An example formal context, the crosses indicating a value of 1 for I, is shown in Figure 2.

Based on this formal context, we build a concept lattice (illustrated in Figure 2) allowing us to identify the sets of interests which are best suited to be taken into account for the definition of user communities. The lattice assigns users to (sets of) interests which are the building blocks for the communities. A user U_i is interested in all concepts and attribute(s) values that are connected to and hierarchically higher in the lattice than the node the user is assigned to.

From the Galois lattice we can automatically infer association rules like the ones shown in Figure 3. For example, rule 1 states that 7 users are interested in restaurants with formal dress code, and 6 users out of these 7 prefer french food. From the association rules, we infer suitable communities, which balance generality and specificity. Communities should not be too general because information of interest for the user must be disseminated as much as possible only to users that are interested in that information. Users do not have to be flooded with all kinds of information. On the other hand, communities should not be too specific, because this leads to a large number of sparsely populated communities. Intuitively, this corresponds to concepts in the lattice on a not too high and not too low level, i.e. somewhere in the middle.

The association rules allow us to identify a balanced number of communities with users who share common interests. Two basic metrics are used to find the sets of interests defining the communities:

- Support s : This metric denotes the proportion of users that expressed their interest in a set of attributes. For example, association rule 4 indicates that 6 users (out of the total of 15) have an interest in restaurants serving French food with formal dress code, meaning support = 0.4.
- Confidence c : This metric expresses the proportion of users having an interest in the rules consequent, given they have an interest in the rules antecedent. In rule 4 for example, 5 out of 6 users that are interested in French restaurants with formal dress code are interested in Parisian restaurants. So confidence in this case is 0.83.

The interest set of a community is defined as the attributes of those association rules which have a minimal support s_{min} and minimal confidence c_{min} . These minimal values can be tuned according to the application needs and priorities. For example, if it is important that not much information is disseminated in

communities that is irrelevant for some users, c_{min} should be put rather high. For our purposes, we set s_{min} to 0.4 and c_{min} to 0.8. Minimal support ensures that only concepts with extents having at least $100 \cdot s_{min} \%$ of all users are considered candidates for the communities. Algorithms for computing frequent item sets are used to build such a so called Iceberg lattice [1].

As soon as the appropriate candidate interest sets for the communities are identified, a second filtering step takes place using the stability index metric ϕ [2].

Let $K = (G, M, I)$ be a formal context and (A, B) be a formal concept of K . Then ϕ is as follows:

$$\phi(A, B) = |\{C \subseteq A \mid C = B\}| / 2^{|A|}$$

Informally, a concept (i.e. community in our case) is considered stable if its intent (set of interests) does not depend much on each particular object of the extent (set of users). Stability expresses how much the set of interests forming a community depends on the interests of its individual users. If a community is persistent, it does not depend on a few users: some users leaving the community should not change the set of interests the community represents. It should be noted that we already minimize this risk by incorporating the minimal confidence metric in step 1. A stable community also does not merge with a different community or split into several independent subcommunities when some users leave the community.

Apart from the *intensional* stability, we also use the *extensional* stability, which measures how the users of a particular community depend on the particular interests. If we remove some interests from the community definition, does the set of users remains more or less the same. Extensional stability indicates how much the community of users depends on a particular interest.

C. Community Recommendation Using Local Views

The previous section explained how communities can be derived given the interests of a set of users by building a global lattice. This lattice gives a global view of the communities and serves as a bootstrap to start users joining and leaving communities according to their needs. When a user found a community that matches his interests and joins it, the members of that community can recommend other possibly interesting communities to the newly joined that they themselves are member of. This collaborative filtering process allows to exploit the local knowledge of each user about communities he is a member of to efficiently recommend other communities.

The local information is represented by local lattices, where the objects are the users and the attributes are communities that the users are member of. The formal context defines a membership function of users to communities for one specific community A , as is exemplified for a community with 4 users in the Figure 4.

When a user U_5 joins community A , it will publish its community membership information and the local lattice is updated. First, the local lattice, shown in Figure 4, is used to check which communities are popular in community A , and communities are accordingly recommended to U_5 . In

	LOC:PARIS14	FT:ITALIAN	FT:INDIAN	...	DC:CASUAL	DC:FORMAL
U_1	X		X			X
U_2	X	X	X		X	
U_3		X	X	X	X	
...				X	X	
U_I	X	X			X	
...		X	X		X	
U_N	X			X		

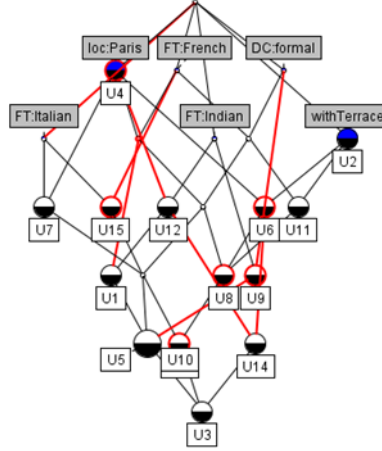


Fig. 2. Formal context and resulting lattice

1	< 7 >	DC:formal=[86 %]	=>	< 6 >	FT:French;
2	< 7 >	FT:Indian=[86 %]	=>	< 6 >	loc:Paris;
3	< 5 >	loc:Paris DC:formal=[100 %]	=>	< 5 >	FT:French;
4	< 6 >	FT:French DC:formal=[83 %]	=>	< 5 >	loc:Paris;
5	< 6 >	loc:Paris FT:French=[83 %]	=>	< 5 >	DC:formal;
6	< 6 >	FT:Italian=[83 %]	=>	< 5 >	FT:French;
7	< 6 >	FT:Italian=[83 %]	=>	< 5 >	loc:Paris;
8	< 4 >	loc:Paris FT:Italian FT:French=[100 %]	=>	< 4 >	DC:formal;
9	< 4 >	FT:Italian DC:formal=[100 %]	=>	< 4 >	loc:Paris FT:French;
10	< 4 >	FT:Indian FT:French=[100 %]	=>	< 4 >	loc:Paris;
11	< 4 >	FT:French withTerrace=[100 %]	=>	< 4 >	DC:formal;
12	< 4 >	DC:formal withTerrace=[100 %]	=>	< 4 >	FT:French;
13	< 5 >	conn:wifi=[80 %]	=>	< 4 >	FT:Indian;
14	< 5 >	loc:Paris FT:French DC:formal=[80 %]	=>	< 4 >	FT:Italian;
15	< 5 >	FT:Italian FT:French=[80 %]	=>	< 4 >	loc:Paris DC:formal;
16	< 3 >	loc:Paris FT:Indian FT:French DC:formal=[100 %]	=>	< 3 >	FT:Italian;

Fig. 3. Association rules for identifying communities

our example, the support for community B is 0.75 and this community is recommended to U5. In case U5 is already a member of one or more of the communities in the local lattice, a recommendation will be made containing the direct children of these communities.

V. RELATED WORK

Search services on mobile devices are less explored and mature than their desktop counterparts. However, previous research related to web search enhancement and personalization on the web can be extended to mobile devices.

The representation of the domain and communication knowledge about users and contextual conditions involved in the retrieval process, have enabled significant improvements in the performance of retrieval services including:

- Personalized content search and browsing based on user preferences;
- Semantic representation of user preferences [3];
- Dynamic contextualization of user preferences;
- Dynamic social networking and collaborative filtering.

Recent approaches to semantic representation of user profiles (research interests) is based on ontology-based representa-

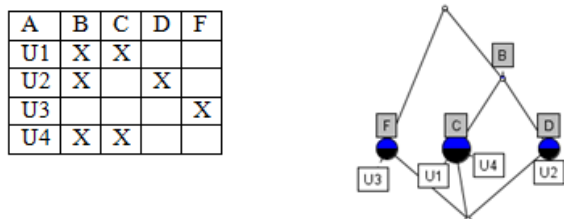


Fig. 4. Local lattice for community A

tions and semantic descriptions that provide new opportunities for improvement of personalization from a richer representational level [4] [5]. In ontology-based approaches, semantic user preferences may be represented as a vector of weights referring to the intensity of the users' interest for each concept.

In addition, most of these approaches propose to generate a matrix representation using the search history and domain ontologies. This representation can be used to generate concept lattice from unit objects and consequently to cluster them using formal concept analysis. Indeed, there was some interest in the use of lattices for information retrieval [6]. These approaches propose to build the concept lattice associated with a document/term relation and then to cluster documents according to the user query. Therefore, the main idea in this work is to explore the use of formal concept analysis to enhance user clustering, rather than document clustering, with the aim of applying collaborative filtering in recommendation tasks.

However, in case of huge size of the domain ontology, it is difficult to structure the interest and limit its scope in the light of a fragment of ontology rather than the entire domain ontology. Therefore, ontology modularity can be suitable to tackle these problems. Modularization, in its generic meaning, denotes the possibility to perceive a large knowledge repository (an ontology) as a set of modules, i.e. smaller repositories that, in some way, are parts of and compose the whole knowledge [7]. Ontology modularization is defined as a way to structure ontologies, so that large domain ontologies will be the aggregation of self-contained, independent and reusable knowledge components (considered as Ontology module (OM) [8].

On the other hand, some others approaches are focusing on using ontologies for creating clusters of results. The application of concept lattices in information retrieval was originally adapted to this task [9]. One limitation is the complexity of the lattice (regarding the size and the needed computation) for large contexts. Therefore, ontology modularity and formal concept analysis on ontology modules enable to solve this problem.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a novel approach to efficiently represent dynamically evolving user preferences and interests. By analysing the dialogue history of the user, interests are inferred and ontology modules for different domains are

annotated with scores. In this way, our spoken dialogue system can provide personalized answers to the users search goals. Second, the interests are used to perform formal concept analysis and to construct ad-hoc communities of users sharing similar interests. By collaborative filtering we can share and recommend possibly interesting information *and* additional communities to users.

For future research directions, we consider first of all to integrate the approach in the PARLANCE spoken dialogue system. A practical evaluation of the proposed lattice-based community building mechanism will allow us to analyze performance of the proposed approach, and to demonstrate the feasibility of this approach, mainly through a formal analysis of the complexity building lattices and inferring association rules.

REFERENCES

- [1] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal, "Computing iceberg concept lattices with titanic," *Data Knowl. Eng.*, vol. 42, no. 2, pp. 189–222, Aug. 2002. [Online]. Available: [http://dx.doi.org/10.1016/S0169-023X\(02\)00057-5](http://dx.doi.org/10.1016/S0169-023X(02)00057-5)
- [2] S. Kuznetsov, S. Obiedkov, and C. Roth, "Reducing the representation complexity of lattice-based taxonomies," in *Proc. of ICCS 15th Intl Conf Conceptual Structures*, ser. LNCS/LNAI, U. Priss, S. Polovina, and R. Hill, Eds., vol. 4604. Springer, 2007, pp. 241–254.
- [3] A. Sieg, B. Mobasher, and R. Burke, "Ontological User Profiles for Personalized Web Search." [Online]. Available: <http://maya.cs.depaul.edu/~mobasher/papers/arch-itwp07.pdf>
- [4] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, and M. G. Strintzis, "Semantic annotation of images and videos for multimedia analysis," in *The Semantic Web: Research and Applications: Proceedings of the Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29-June 1, 2005*, ser. Lecture Notes in Computer Science, A. Gómez-Pérez and J. Euzenat, Eds., vol. 3532. Springer, Berlin–Heidelberg, Germany, 2005, pp. 592–607.
- [5] P. Castells, M. Fernández, D. Vallet, P. Mylonas, and Y. Avrithis, *Self-tuning personalized information retrieval in an ontology-based framework*. Springer-Verlag, 2005, vol. 3762, pp. 977–986. [Online]. Available: <http://www.springerlink.com/content/64mfxx49m3ga019>
- [6] C. Carpineto, G. Romano, F. U. Bordoni, and F. U. Bordoni, "Exploiting the potential of concept lattices for information retrieval with credo," *Journal of Universal Computer Science*, vol. 10, pp. 985–1013, 2004.
- [7] *Modularization: a Key for the Dynamic Selection of Relevant Knowledge Components*, 2006. [Online]. Available: <http://people.kmi.open.ac.uk/mathieu/papers/womo06.pdf>
- [8] H. Stuckenschmidt, C. Parent, and S. Spaccapietra, Eds., *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2009, vol. 5445.
- [9] J. Khler, S. Philippi, and M. Lange, "Sameda: ontology based semantic integration of biological databases," *Bioinformatics*, vol. 19, no. 18, pp. 2420–2427, Dec. 2003.