

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

## An ontology-based approach for personalized RESTful Web service discovery

Sana Ben Abdallah Ben Lamine<sup>a</sup>, Hajar Baazaoui Zghal<sup>a</sup>, Michael Mrissa<sup>b</sup>, Chirine Ghedira Guegan<sup>c\*</sup>

<sup>a</sup>RIADI Laboratory, University of Manouba, Manouba, Tunisia

<sup>b</sup>LIUPPA Laboratory, University of Pau and Adour Countries, Bayonne, France

<sup>c</sup>Université de Lyon, CNRS, IAE, University Lyon 3, LIRIS, UMR5205, France

---

### Abstract

Web service discovery is a challenging task that has received a lot of interest in the last decade. Several approaches have been proposed, however, many limitations remain. In this paper, we focus on collaborative semantic discovery of RESTful services according to the HATEOAS principle. We propose an approach based on 1) semantic links between services; 2) user profile clustering to dynamically measure user similarity for queries in a similar context, and 3) a user profile ontology to manage users and corresponding services. A prototype has been implemented to evaluate our proposal and show its effectiveness.

© 2017 The Authors. Published by Elsevier B.V.  
Peer-review under responsibility of KES International

*Keywords:* RESTful Web services; linked services; user profile ontology

---

### 1. Introduction

With the exponential growth of the number of Web services published on the Web, users face an increasing difficulty to discover the most appropriate services to answer their requests. This fact leads to a need for solutions to enable users to discover the most adequate service, which ensures the highest satisfaction, within a specific user context.

---

\* Sana Ben Abdallah Ben Lamine, Hajar Baazaoui Zghal, Michael Mrissa, Chirine Ghedira Guegan.

E-mail addresses: [sana.benabdallah@riadi.rnu.tn](mailto:sana.benabdallah@riadi.rnu.tn), [hajar.baazaouizghal@riadi.rnu.tn](mailto:hajar.baazaouizghal@riadi.rnu.tn), [michael.mrissa@univ-pau.fr](mailto:michael.mrissa@univ-pau.fr), [cghedira@liris.cnrs.fr](mailto:cghedira@liris.cnrs.fr)

Several studies have focused on the personalization of web services requests from a user and of responses to these queries, based on past users' experiences <sup>1, 2, 3, 4, 5, 6, 7</sup>. Therefore, measuring the degree of user satisfaction for each service invocation would help refine the discovery of resources returned as response to a given query. Yet, we think that the results would be further improved with a search in service collections approved by a community of users with whom the user emitting the query shares similar opinion concerning a set of resources used in similar contexts. Indeed, search results will benefit from the previous users' experiences in a specific context and will enable prediction of the user's satisfaction for each resource from the users' neighbors. In this paper, we focus on RESTful Web services, which are becoming increasingly used for the development of resource-oriented architecture (ROA) or data oriented architecture (DOA). Unlike classical Web services, RESTful-based ones offer a uniform resource-oriented interface and comply with the Hypermedia As The Engine Of Application State (HATEOAS) principle that promotes the use of hypermedia links to navigate between the states of an application. To this end, we propose a solution that allows to describe and discover RESTful Web services based on semantic user profiling techniques. Service discovery is personalized regarding to the user's context and feedbacks and those of his similar users.

The scientific contribution of the work described in this paper consists in the following points: (1) HATEOAS-based personalization of RESTful Web service discovery (2) user profile management based on a profile ontology and (3) collaborative filtering for service recommendation.

The rest of this paper is organized as follows: in section 2 we give an overview of existing work related to Web service discovery. Our proposal is introduced and detailed in section 3. Section 4 presents and discusses evaluation and experimental results of our approach. We conclude and present our future work in section 5.

## 2. Background, related work and motivation

In this section, we briefly present RESTful services and linked services and give an overview of related work.

### 2.1. RESTful services

In our work, we deal with RESTful Web services. These latter follow the Representational State Transfer (REST) architectural style for distributed hypermedia systems that is a hybrid style derived from several network-based architectural styles <sup>8</sup>.

Unlike classical Web services, RESTful-based ones allow message sending without any SOAP like envelope and directly use typical encoding (XML, JSON, binary, text). They also offer a uniform resource-oriented interface that allows similar interactions with any RESTful client, thus enabling the use of a generic client. Another interesting feature of REST architectural style is the HATEOAS principle that promotes the use of hypermedia links to drive the evolution of applications.

Although REST is not a standard, it is becoming more used for the development of resource-oriented architecture (ROA) or data oriented architecture (DOA). In our work, a resource is any piece of information that can be accessible through its URI, in other words, any concept that might be the target of an author's hypertext reference <sup>8</sup>.

In this paper, we consider a RESTful Web service as a set of resources that provide a coherent access to the state and functionality of the software it represents <sup>9</sup>.

### 2.2. Linked services

Linked services, introduced around 2010 <sup>10</sup>, are Web services that are described by linked data, and that exchange linked data too. These services combine the advantages of the principles followed by service-oriented computing (loose coupling, composability...) with those of linked data (use of semantic descriptions to enable automatic processing). Linked services are a promising way to automate the discovery, composition and execution tasks, which have been proven to be challenging in typical service-oriented architectures.

### 2.3. Related work

As aforementioned, our proposal aims to improve the discovery of linked RESTful services with the help of collaborative filtering techniques. It is based first on the user experience and his feedback, and secondly, on the former experience of a set of users deemed similar thereto.

Hence, we identified related works that deal with RESTful service description and discovery, described in the following.

In <sup>11</sup>, a social model provides the means to link services to each other. The relationships between services drive the discovery algorithm, which remains based on a centralized setup.

Later work <sup>12</sup> proposes to discover RESTful service with a slightly modified social model and according to the HATEOAS principle. We relied on this approach to build the discovery approach presented here. Other work proposes semantically annotated descriptions to support RESTful service discovery, as microformats <sup>13</sup> or as full semantic descriptions <sup>14</sup>, with some work explaining how modus ponens inference can drive the discovery process<sup>15</sup>.

Another category of work relates mainly to service discovery based on user experience.

In <sup>1, 2, 3</sup> the customization of services is based on quality of services. In <sup>1</sup>, the authors present WSMO-QoS which refines the non-functional properties class in WSMO, the web services description language. A normalization algorithm for the selection of a service is used. In <sup>2</sup>, the authors propose OWL-Q as an extension for OWL-S and a rule-based algorithm to match two metrics. In <sup>3</sup>, the authors propose to use an ontology for the specification of non-functional properties (NFPs) to customize the discovery of semantic web services. The selection is based on declarative rules to perform the logical matching between the service NFPs and those requested.

Other work <sup>4, 5, 6</sup> takes into account the user's search context throughout the access process to the resource. In <sup>4</sup>, the authors define the context as "the set of conditions and environmental influences that make the situation unique and make it possible to understand it". In <sup>5</sup>, the web services are selected by matching their contexts to that of the user. The context elements that are represented in XML format are defined as pairs (name, value), the matching of the parameters is based on logical equivalence whereas the matching of the values is based on the inclusion. In <sup>6</sup>, the context is represented by a generic ontology that can connect several domain ontologies. Matching is based on the semantic distance.

Other work focuses on the user model defined by <sup>7</sup> as "a knowledge source, a database on a user." This knowledge about the user can be identity, geolocation, interests, or preferences. In <sup>16, 17</sup>, the vector model is used to represent user information. In <sup>18</sup>, the authors have constructed an ontology describing user profiles for the discovery of web services as part of an e-learning system. The profile can be organized in several dimensions which can be of different formalisms. Each dimension describes a facet that represents the user <sup>19</sup>. In <sup>20</sup>, authors proposed a customized information retrieval system whose proposed user profile is both semantic and multidimensional.

Despite these numerous works, several limitations have been noted. Traditional service-oriented architectures support service discovery with a central registry (i.e. UDDI) that contains all the meta-data necessary to describe available services. Typical service composition algorithms (planning-based, etc.) apply to such a setup, which has been successful in closed-world environments such as small or medium-size organizations. On the contrary, the REST architectural style envisions service discovery through hypermedia and self-describing messages, thus building distributed hypermedia applications that progressively follow semantically explicit hypermedia links to drive the application state (as per the HATEOAS principle of REST). Hence, as no central registry is available to answer a request, typical composition algorithms do not apply to distributed hypermedia applications. The discovery algorithms that drive service composition need to be re-designed. In addition, although we know user profile matching techniques and collaborative filtering have been applied to service discovery approaches based on a centralized registry, HATEOAS-based discovery approaches are mostly recent. To the best of our knowledge, no existing work has validated that they could be combined with user profile matching techniques and collaborative filtering in a relevant way. Given that, we aim in this paper to demonstrate that combining HATEOAS-driven discovery with user profile matching is an interesting solution to drive the discovery of RESTful services. Indeed, we propose an approach based on semantic links between services and user profile clustering to dynamically measure user similarity for queries in a similar context. In addition, a user profile ontology is defined to manage users and corresponding services.

### 3. Ontology-based approach for personalized RESTful Web service discovery

In this work, we propose an ontology-based approach for personalized RESTful Web service discovery. We aim at personalizing the results returned to different users discovering RESTful Web services. The personalization is based on the use of a profile ontology, which saves for each user, his similar users and his formerly used services with the corresponding feedback. A services' relationship ontology is utilized to highlight similarity and complementarity relationships between services. The collaborative filtering method used by the recommendation systems is used to filter returned services with respect to a predicted utility (or satisfaction) value of the returned service, based on the opinions of similar users.

Our proposal is based on the general architecture depicted in figure 1 that falls into two layers.

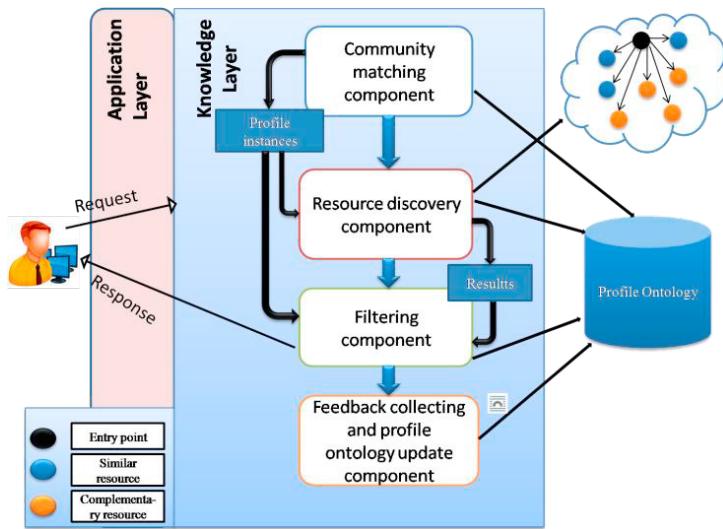


Fig. 1. Proposed general architecture.

The application layer contains the services provided to the user. The user interface is firstly displayed, it allows the identification that enables the creation of a new model for new users or load the model for registered ones. The search process begins when the user submits a query that allows two types of search: a classic search in the case of a new user or a personalized search in the case of an identified user. The search results are then exposed to the user after a treatment by the knowledge layer.

The knowledge layer is composed of four components: (1) Community matching component, (2) Resource discovering component, (3) Filtering component and (4) Feedback collecting and profile ontology update component.

The latter components are detailed in the following subsections.

#### 3.1. Community matching component

This component aims to determine a set of similar user circles for the current user. Two scenarios are possible in the community matching component's algorithm: (1) The user is identified and matched with the appropriate community of users, by searching in the profile ontology (defined in Definition 1 and presented in Fig. 2) for users who have previously used services similar to that requested by the current user and in a similar context and (2) for a new user, a new profile is created in the application layer.

Definition 1: A profile Ontology for a user  $u$  is defined as follows:

$$O_u = \{C, R, A\} \quad (1)$$

Where C is a set of concepts, R is a set of relations and A is a set of axioms expressed in a logical language.

The profile ontology  $O_u$  represents knowledge about user and covers the main aspects of the user's profile (cf. Fig. 2 for more details).

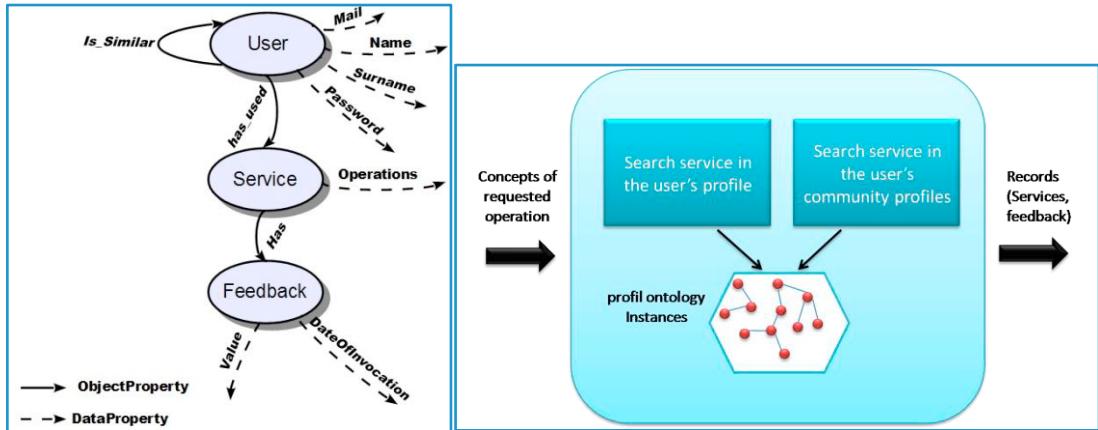


Fig. 2. Proposed profile ontology

Fig. 3. Resource discovering component for an identified user

The community matching algorithm's inputs and outputs are:

**Input:** user **U**, profile Ontologie **O**

**Output:** [Profiles of Similar Users **psim**]

### 3.2. Resource discovering component

To discover services, two scenarios are foreseen (1) Identified user and (2) New user or limited experience:

(1) Identified user (cf. Fig.3): the user is identified in the application layer and the approach can take advantage of the user's historic as well as his group of similar users for the requested service. The user's profile is used during the discovery process. The approach refers to the user's experience and questions the profiles ontology described in Fig. 2 to extract the services that (a) are already evaluated by the user himself (cf. line 2 algorithm 1) or by members of the same community (cf. lines 3 to 5 algorithm 1). and (b) that meet the need of the given user (cf. algorithm 2).

Typically, a user asks the Web service search engine to request an operation. The query is processed to extract the concepts, using the domain ontology described in <sup>12</sup>. The search of services that meet the identified concepts is accomplished by establishing correspondence between them and service annotations in users' profiles.

(2) New user or limited experience: In such case, the approach performs a classical search exploring, from an input point (cf. Fig. 4), the related resources and discovers those meeting the user's need, in accordance with the approach proposed by <sup>12</sup> for resources discovery by exploring its descriptors (cf. line 7 algorithm 1).

In fact, as it has been stated in the state of the art, REST is an architectural style that supports the notion of hypermedia. Each service is linked to a set of services, which favors HATEOAS. In <sup>12</sup>, two types of relationships between web services are proposed: "Is-Similar" and "Is-Complementary". Two services are similar if they have the same functionalities (operations) and two services are complementary if one of the different functions (operations) is of interest to the user, for example a train ticket reservation service and reservation of hotel room service are complementary.

The semantic description proposed by <sup>12</sup> allows us to consider the Web as a graph where nodes are services and edges are the semantic links between them. Moreover, semantic annotation links between the services: "Is-Similar" and "Is-complementary" allows us to navigate in the graph (cf. Fig. 4).

Algorithms 1 and 2 detail the resource-discovering component.

---

**Algorithm 1** Personalized service discovering Algorithm
 

---

**Input:** Operation-concept C, [Profiles of SimilarUsers **psim**], User-Profil **P**, Entry-point **E**  
**Output:** {records (Service, Feed)} **R**

- 1:**Begin**
- 2:  $R \leftarrow \text{Search-Services-In-Profil} (P, C)$
- 3: **ForEach**  $P_i$  in **PSim** **do**
- 4:  $R \leftarrow R + \text{Search-Services-In- Profil}(P_i, C)$
- 5: **end ForEach**
- 6: **if** (empty (R)) **then**
- 7:  $R \leftarrow \text{SearchInWeb} (E, C)$
- 8: **endif**
- 9: **End**

---

**Algorithm 2** Search-Services-In-Profil (P,C)
 

---

**Input:** User profil **P**, Operation concept **C**  
**Output:** {records (Service, Feed)} **R**

- 1: **Begin**
- 2: **ForEachService**  $S$  in **P**
- 3: **if** (ServiceOperationHas ( C ) **then**
- 4:  $F \leftarrow \text{GetFeedback}(S, P)$
- 5: **if** (systemDate -  $F$ .dateOfInvocation < threshold) **then**
- 6:  $R \leftarrow R + \text{Record} (S, F)$
- 7: **endif**
- 8: **endif**
- 9: **endFor**
- 10: **End**

---

**Algorithm 3** Filtering Algorithm
 

---

**Input:** {record (S,F,U)  $R$ } services found in Similar Users' profils and the value of their feedback, {record (SU,FU)  $R_u$ } services found in User's profil and their feedback

**Output:**  $S_1 = [s_1, s_2, \dots]$  set of resources (services) to recommend, sorted by satisfaction prediction.

- 1:**Begin**
- 2:**ForEach**  $R_i$  in **R** **do**
- 3:  $S_i \leftarrow R_i.S$
- 4: **if** ( $R_i.F \leq 2$ ) **then**
- 5:     EliminateService ( $S_i, R, R_u$ )
- 6: **endif**
- 7: **endFor**
- 8:  $S \leftarrow S + R_i.S$
- 9:  $j = 0$
- 10: **ForEach**  $i$  in **S** **do**
- 11: **if** (exists ( $i, R_u$ )) **then**
- 12:      $P(r_{u,i}) = R_i.F$
- 13: **else**
- 14:      $U_a = R_i.U$

$$P(r_{u,i}) = \bar{u} + \frac{\sum_{u_a \in S(u)} \text{Sim}'(u_a, u)(r_{u_a,i} - \bar{u}_a)}{\sum_{u_a \in S(u)} \text{Sim}'(u_a, u)}$$

- 15:
- 16: **endIf**
- 17:  $\text{Predictions}[j] \leftarrow P(r_{u,i})$
- 18:  $j \leftarrow j + 1$
- 19: **endFor**
- 20:  $\text{Sort} (S, \text{Predictions})$
- 21: **End**

---

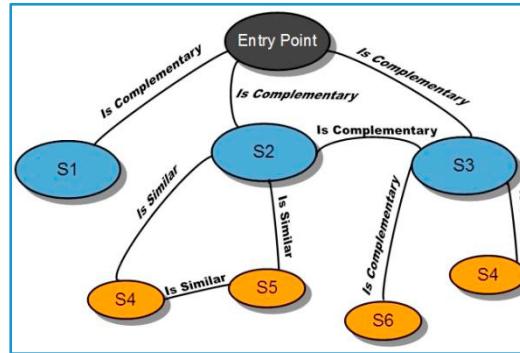


Fig. 4. Relationship between RESTful Web services

### 3.3. Filtering component

The resource discovery phase can return a large number of services that are similar and thus all respond to the user's request. Even if all services are similar and have all the same features, they can bring uneven levels of satisfaction to each user. For this reason, we propose to follow the discovery phase by a filtering, based on feedback from the user and feedback from similar users, in order to return the most relevant services which can bring the highest level of satisfaction.

The need to rely on feedback from similar users is especially necessary if a user does not have previous experiences with a returned service, which makes the prediction of its relevance to the user very difficult. The filtering component (cf. Fig. 5) is based on the collaborative filtering method used by the recommendation systems. The filtering algorithm is detailed in algorithm 3.

For each service returned by the discovery phase, we calculate a utility value which indicates a satisfaction prediction of the service to the user. This latter value is calculated using the  $P(r_u, i)$  function used in collaborative filtering systems (cf. line 15 algorithm 3) based on the opinions of similar users, scanning the instances of the profile ontology returned by the resource discovery phase (component 1). We notice that returned services with a satisfaction predicted value less than 2 are eliminated and considered as not interesting to the user in question (cf. lines 2 to 7 Algorithm 3).

The output of the filtering phase is a set of services to recommend to the user, sorted by the calculated satisfaction prediction value (cf. line 20 algorithm 3).

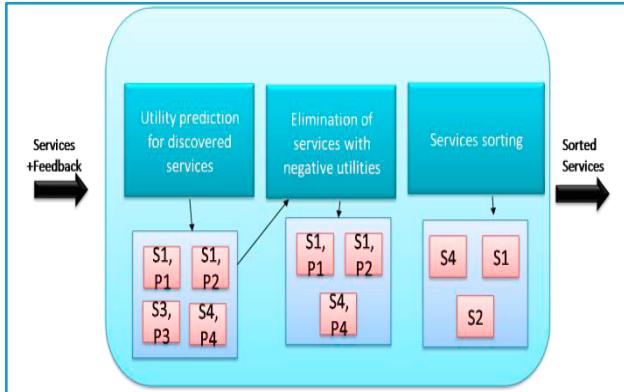


Fig. 5. Filtering component

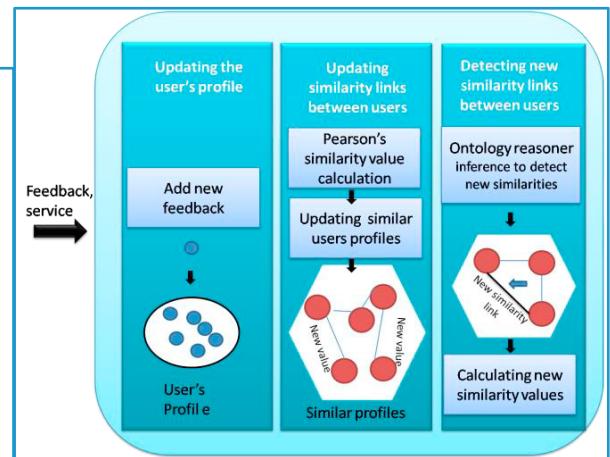


Fig. 6. Feedback collecting and profile ontology update component

### 3.4. Feedback collecting and profile ontology update component

Having the results given by the Filtering component, the user will be able to choose a service and to invoke it. The next phase is accomplished by the component 4 (cf. Fig. 6), which is responsible for the feedback collecting and for the profile ontology update.

The functioning of the component 4 begins then by (1) Updating the user's profile: First, the user feedback concerning the used service, is collected. In fact, after a fixed period of use of this service, the user is asked for his feedback about it. The user affects a satisfaction mark between 0 and 5 (from not interesting to very interesting services). Second, to finalize the user's profile updating step, the appropriate feedback is added to the ontology. The component 4 continues then by (2) updating similarity links between users: the similarity between users is based primarily on services assessments. In our work, we use Pearson's correlation, used in collaborative filtering systems to measure similarity between users. Therefore, the Pearson's similarity value is computed between the current user and his group of similar users who already have an assessment for the invoked service. Finally, the system shifts into (3) detecting new similarity links between users: we define a transitive "is similar" relationship between two users.

The ontology reasoner will be able to infer new relationships between users (A "is similar" with B and B "is similar" with C then A "is similar" with C). This method allows us to restrict the set in which the search for new similarity relationships is done. Then, the similarity degree between the current user and the new ones inferred by the ontology's reasoner is computed.

#### 4. Implementation and experimental evaluation

In order to evaluate the effectiveness of our proposal, we developed an application using J2EE technology. The proposed application provides a user interface and supports the personalization process described in previous sections. It also supports Web service retrieval in order to capture user interaction (relevance and activity indicators). The purpose of this section is to disclose how that can be achieved when a query is submitted using our application. We demonstrate how the returned results are personalized for each different user.

##### 4.1. Implementation

From a technical point of view the technologies that have been employed in the implementation of our proposal are the following:

- J2EE which is Oracle's enterprise Java computing platform (<http://www.oracle.com>);
- OWL API to submit queries to Google search engine and obtain a set of Web pages as a result (<http://owlapi.sourceforge.net>);
- Protégé that allows the edition and visualization of ontologies (<http://protege.stanford.edu>);
- Hermit OWL Reasoner (Hermit OWL Reasoner)

The implementation allowed us to conduct a series of evaluations that measure the performance of different scenarios and to compare the obtained results.

##### 4.2. Experimental setup

To experiment our approach, we choose the “buying book domain” described in Fig. 7. Users look for services like buying books, achieving payment, search for shipping methods, etc. The evaluation process is applied on a collection that contains 50 services with 32 services belonging to our retrieval domain “buying book”. The remaining services belong to other domains like tourism, weather, etc.

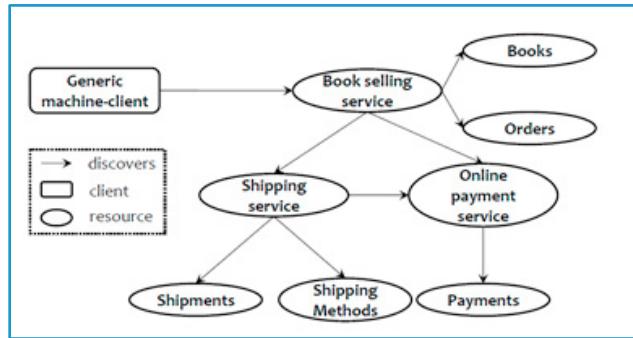


Fig. 7. Buying book domain description

In addition to the RESTful Web services collection, a profile collection composed of 30 users' profiles is used. The profiles used are those of the members of our laboratory who helped us to validate our proposal.

##### 4.3. Evaluation of returned results

To evaluate the results returned by our approach, we first measured the recall. We varied the number of users in the profile' collection, from 5 to 30 (with a pitch of 5) (cf. Fig. 8).

Fig. 8 demonstrates an enhancement of the recall as the number of profiles is incremented. This is due to the augmentation of the number of similar users in the collection, which helps enhancing the user experience.

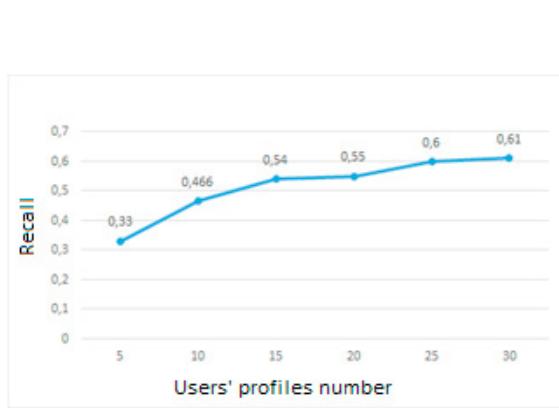


Fig. 8. Variation recall value according to the number of users' profiles



Fig. 9. Variation of predicted utility values of the different services according to different users

The results improvement and personalization is secondly demonstrated by executing the same request  $R = \text{"ConsultBook"}$  by a sample  $U = [u_1 \dots u_{10}]$  of 10 users represented by instances in the profile ontology presented in Fig. 2 and having an experience with our collection of services.

In the latter collection, six services  $S_0 \dots S_5$  have operations annotated by the requested concept.

Table 1 summarizes the predicted utility values calculated by the filtering component (cf. Fig. 5) for the 5 services to each of the 10 users.

Analyzing table 1, we first notice some empty boxes that indicate that the corresponding service was eliminated in the filtering phase (cf. Fig. 5) because services with a satisfaction predicted value less than 2 is eliminated and considered as not interesting to the user in question.

Table 1. Variation of the predicted utility value of the services according to different users' profiles.

Users \ Service	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
U1	3.44		3			3.43
U2		3		3.03		3.46
U3	3			2.83		2.79
U4	2.3	2.33	3			4
U5	3.89		3.38	3.65		3.42
U6	3.34	3.13		2.88		
U7		2.66	2.59			
U8	4		3.9			
U9	3	3.07	2.59	4		
U10	3	2.66				4

The bar chart in Fig 9. represents the results summarized in Table 1. Analyzing Fig.9, it can be noted that the same request sent to our system from ten different Users ( $U_0, \dots, U_{10}$ ), returns different predicted utility values concerning the five services ( $S_0, \dots, S_5$ ) (e.g.  $U_{10}/S_5: 4$ , while  $U_1/S_5: 3.43\dots$ ) which demonstrates the personalization of the returned results. In fact, concerning the same request, results are different with respect to the users' profiles, their previous experience and feedback concerning the invoked services. The similar users' profiles are also considered.

The increase in the number of users and the services should give the same result expected or better since there will be more opinion to refine the discovery.

## 5. Conclusions and future work

In this paper, we have outlined an ontology-based approach for personalized RESTful Web service discovery. The proposed general architecture is organized into two layers: the application layer which contains the services

provided to the user and the knowledge layer, which is composed of four components: (1) a community matching component, (2) a resource discovering component, (3) a filtering component and (4) a feedback collecting and profile ontology update component.

The originality of the work described in this paper consists of the following points: (1) personalization of RESTful Web service discovery based on HATEOAS, (2) user profile management based on a profile ontology highlighting users experience with the services by capturing their feedback and those of similar users, and (3) collaborative filtering to recommend services based on their utility to the users.

A prototype has been developed on the Java Platform. Experiments and evaluations have been carried out, which highlight that overall achieved improvement are obtained thanks to the integration of ontologies and users' profile management.

In future work, we aim at studying how the use of fuzzy ontologies into the process can help take into account the uncertainty in the similarity of users and services. Scalability issues are also to be examined in the future.

### Acknowledgements

This work was financially supported by the “PHC Utique” program of the French Ministry of Foreign Affairs and Ministry of higher education and research and the Tunisian Ministry of higher education and scientific research in the CMCU project number 17G1122.

### References

1. Wang X, Vitvar T, Kerrigan M, TomaI. A QoS-aware selection model for semantic web services., Proc. International Conference on Service-Oriented Computing (ICSO 2006); 2006, pp: 390-401, doi: 10.1007/11948148\_32.
2. Kritikos K, Plexousakis D. OWL-Q for semantic QoS-based web service description and discovery. Proc. International Conference on Service Matchmaking and Resource Retrieval in the Semantic Web (SMRR 2007), CEUR-WS.org, 2007; Vol. 243, pp. 114–128.
3. Ouhab A, Malki M. Découverte personnalisée de services Web sémantiques.CIIA; 2009.
4. Brézillon P, Pomerol J-C. Some comments about knowledge and context. Research Report 2001-022, LIP6, Université Paris VI, Paris, France.
5. Mussi E. Flexible and context-aware processes in service oriented computing. Phd thesis, Dipartimento di Elettronica e Informazione, Vol. Ph.D., Politecnico di Milano; 2007.
6. Rerrer-Brusch U. Service matching with contextualized ontologies. Phd. Thesis, Paderborn; 2006.
7. McTear M F. User modelling for adaptive computer systems: a survey of recent developments. Artificial intelligence review; 1993, 7(3-4) :157–184.
8. Fielding R T. Architectural styles and the design of network-based software architectures. PhD. thesis, University of California, Irvine; 2000.
9. Pautasso C. RESTful Web service composition with BPEL for REST. Data Knowl. Eng.;2009, 68(9):851–866.
10. Lanthaler M, Guetl C. Towards a RESTful service ecosystem.In Hussain, F.H. and Chang, E. (eds), Proc. IEEE International Conference on Digital Ecosystems and Technologies (DEST 2010); 2010, pp. 209-214. Dubai, United Arab Emirates: IEEE Computer Society.
11. Maamar Z, Wives LK, Badr Y, Elnaffar S, Boukadi K, Faci N. Linkedws: A novel web services discovery model based on the metaphor of “social networks”. Simulation Modelling Practice and Theory; 2011,19(1):121–132.
12. Bennara M, Mrissa M, Amghar Y. An approach for composing RESTful linked services on the Web. Proc. International Conference on World Wide Web (WWW 2014); 2014, pp. 977–982. ACM.
13. John D, Rajasree M S. RESTDoc: Describe, Discover and Compose RESTful Semantic Web Services using Annotated Documentations. International Journal of Web & Semantic Technology (IJWesT);2013, 4(1).
14. Salvadori L, Siqueira F. A Framework for Semantic Description of RESTful Web APIs. IEEE International Conference on Web Services (ICWS 2014); 2014, vol. 00, no. , pp. 630-637, doi:10.1109/ICWS.2014.93
15. Verborgh R, Steiner T, Deursen D V, Roo J D, de Walle R V, Vallès J G. Description and Interaction of RESTful Services for Automatic Discovery and Execution. Proc. FTRA 2011 International Workshop on Advanced Future Multimedia Services (AFMS 11); 2011.
16. Balabanovic M, Shoham Y. Fab : content-based, collaborative recommendation. Communications of the ACM; 1997, 40(3) :66–72.
17. Oard DW, Kim J. Modeling information content using observable behaviour. Proc. Annual Conference of the American Society for Information Science and Technology (ASIST 2001); 2001, pp. 481-488.
18. Adafer F, Balla A, Amrouche H. L'évaluation de l'apprenant dans les environnements d'apprentissage : problématique et réflexions, International Journal of Information sciences for decision making; second term 2006, ISSN:1265-499X.
19. Tamine-Lechani L, Boughanem M, Zemirli M. Exploiting multi-evidence from multiple user's interests to personalizing information retrieval., Proc. IEEE International Conference on Digital Information Management (ICDIM 2007); 2007, Vol.1, pp. 7–12.
20. Besbes G, Zghal HB, Ghézala HB. Fuzzy ontology-based system for personalized information retrieval. Proc. International Conference Knowledge Discovery and Information Retrieval (KDIR 2014); 2014, pp. 286–293.