Cooperative Discovery for User- centered Web Service Provisioning

Wolf-Tilo Balke
Computer Science Department
University of California
Berkeley, CA, USA
balke@eecs.berkeley.edu

Matthias Wagner
Future Networking Lab
DoCoMo Communications Laboratories Europe
Munich, Germany
wagner@docomolab-euro.com

Abstract

In this paper we present means of personalization for user-centered Web services provisioning. We focus on the different phases of interaction with services and show how a cooperative discovery algorithm can essentially improve the service provisioning. An ontology-based algorithm is presented that makes use of user preferences, differing conceptions, as well as standard usage patterns. With a running example we study how to increase Web services' usability and the overall quality of service.

Keywords: Web Services discovery, Cooperative Service Provisioning, User-centered Services, Semantic Web

1. Introduction

In times of the ubiquitous Internet the Web service paradigm is expected to substantially alter the world of modern business processes. Essential components of this emerging service paradigm are Internet-based, modular applications that provide standard interfaces and communication protocols for efficient and effective service provisioning between different business units or even businesses and customers. Web service interfaces and bindings are defined, described and discovered by XML artifacts, supporting XML message-based interoperability with other services and applications via protocols like SOAP [14]. By now Web services have started to show their usefulness in a wide variety of domains.

Basically we have to distinguish two types of Web services: services meant to support business-to-business interaction and those meant for direct user interaction. So far, research in the area of Web services and standardization mainly focuses on the first kind of services with well-defined interfaces and terminologies that define a shared meaning for machine-to-machine interaction. In these environments services can ensure that in a given situation the execution of a service is sensible and well-defined. On the other hand in business-to-consumer interaction we consider services provided for individual customers, which can't be provided through a fully automated process. Here, the human in the loop needs further considerations, e.g. in deciding about the relevance or the value of a specific service. These types of services have to work as a unit towards the user. They are supposed to perform complex tasks and therefore need advanced means of personalization. Taking the example of a business traveler, in [17] we present a case study of personal, user-centered Web services in future mobile communication environments. It is shown how ubiquitous services may accompany a user throughout his daily routine and how complex services like renting cars or finding the next ATM machine – especially when accessed via mobile devices – demand a more thorough support for personalization that cannot yet be achieved by today's techniques.

In [17] we also already argued that the ultimate goal in user-centered and personalized service provisioning has to be the fulfillment of individual user needs expressed as complex tasks which are typically further divided into simpler sub-goals and subsequently matched to different services. Our view of user-centered services is sketched in figure 1: user information – such as profile data and user preferences – is crucial in all phases of service provisioning: service advertising and discovery, service selection and service execution. Whereas we deal with the personalized selection and execution in detail in [2], in this paper we will concentrate on user-centered advertising and discovery of services according to the special needs and preferences of an individual user. In this context, typical usage patterns and enriching service requests with explicitly or implicitly given preferences of single users or user groups can be used for essentially improved Web service matchmaking.

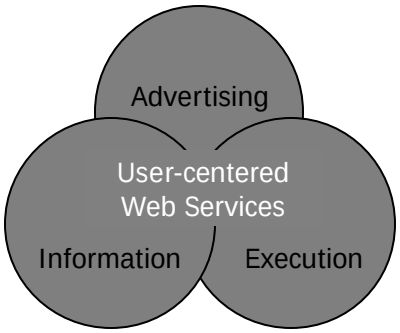

Figure 1: User-centered Web Service provisioning

As a running example throughout this paper (shaded text) we will focus on a sample user interacting with a restaurant booking service. Along the lines of [2] and [17] we will assume that the user discovers booking services of local restaurants with a mobile device and then tries to make the best possible match to select a service for subsequent execution. The running example demonstrates the use of user profiles, usage patterns, and ontologies and proposes adequate techniques like query expansion or ontology relaxation to deal with personalization problems.

2. User-centered Web Services

Initially Web services were mainly intended to engage in dynamic business-to-business interactions with services deployed on behalf of other enterprises or business entities. However, with the evolution of Web service technology networked services will not only become increasingly sophisticated, but also move into the area of business-to-consumer interactions. Internet portals like Yahoo.com already demonstrate the usefulness of a user-centered approach and provide simple personalization mechanisms. For instance, MyYahoo.com allows individual users to define their own portal entry page and determine which personal content to display or which services to use on startup. With Web services becoming an integral component of the future Web, a user-centered approach to personal Web services is easily conceivable, e.g. through Web service technology in the backend of Internet portals.

Trying to model the usage of Web services along the lines on these portal pages, but using the services' full capabilities, interaction with a truly user-centered service can thus be divided into three major phases (see fig. 2):

- the discovery of services that are basically able to perform a task based on a user's service request,
- the querying of services for subsequent selection based on user preferences for deliverable objects,
- and the final execution of a service, after a decision for one of the available objects.

Of course not every Internet service that is deployed as a Web service will be considered user-centered and needs personalization. We basically distinguish between simple services that can always be executed and just return a simple result, e.g. a currency converter or a weather forecast, and complex services, e.g. flight booking or restaurant reservation. Unlike for a currency conversion service that converts say US Dollars to Japanese Yen, for a flight booking service it is not obvious that the execution of the service will deliver the expected result and fulfill a user's needs. Its sensible execution may be dependent on some

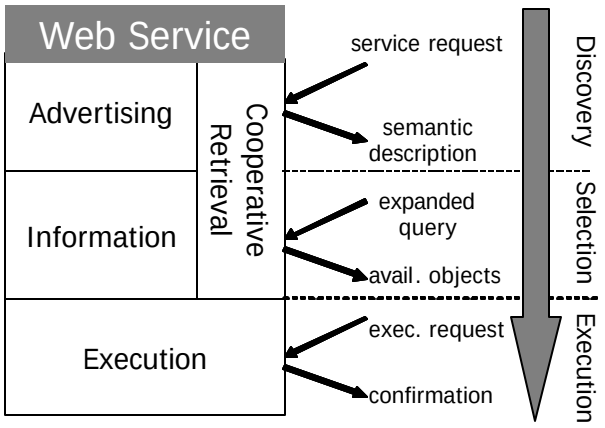

Figure 2: Cooperative Discovery, Selection, Execution

information, e.g. whether a flight with a certain airline can be booked on a certain date. Portals like Expedia.com [7] therefore pose the user's request to a variety of hard-wired services and present the collected results to the user. However, the hard-wiring of the Expedia portal does not allow for the flexibility of the Web services paradigm and thus does not really meet the challenges of the Web.

Interaction	**Tasks**	**Applicability**
business-to-business	well-defined tasks for thorough automation	matching of pre- and post-conditions
business-to-consumer	simple computational tasks	Always
	complex tasks	needing additional information for decision

Table 1: Classifying Web services

Web services thus can be divided into business-to-business and business-to-consumer services and we can further subdivide the latter ones into always applicable services and those whose applicability still depends on a specific piece of information (cf. table 1). From a provisioning point of view the last class of services poses the most problems with respect to personalization, because:

- they usually model complex user-centered tasks consisting of different components like advertising, information and execution (cf. figure 1)
- the final decision which service to execute almost always has to be performed directly by the user, because different services may offer various advantages beyond the specified user preferences.

```
<daml:Class rdf:about="http://myresto.com#Restaurant">
  <rdfs:label xml:lang="en">http://myresto.com#Restaurant</rdfs:label>
   <rdfs:subClassOf
 rdf:resource="http://myresto.com#DEFAULT_ROOT_CONCEPT"/>
  <daml:subClassOf>
    <daml:Restriction daml:minCardinality="1">
      <daml:onProperty rdf:resource="http://myresto.com#foodFlavor"/>
      <daml:toClass rdf:resource="http://myresto.com#Flavor"/>
    </daml:Restriction> </daml:subClassOf>
  <daml:subClassOf>
    <daml:Restriction daml:minCardinality="1" daml:maxCardinality="1">
      <daml:onProperty rdf:resource="http://myresto.com#category"/>
      <daml:toClass rdf:resource="http://myresto.com#Category"/>
    </daml:Restriction> </daml:subClassOf>
</daml:Class>
<daml:Class rdf:about="http://myresto.com#Chinese">
   <rdfs:label xml:lang="en">http://myresto.com#Chinese</rdfs:label>
   <rdfs:subClassOf rdf:resource="http://myresto.com#Asian"/>
</daml:Class>
<daml:Class rdf:about="http://myresto.com#Cantonese">
   <rdfs:label xml:lang="en">http://myresto.com#Cantonese</rdfs:label>
   <rdfs:subClassOf rdf:resource="http://myresto.com#Chinese"/>
</daml:Class>
<a:Cantonese rdf:about="http://myresto.com#Wong_s_Grill">
   <rdfs:label xml:lang="en">http://myresto.com#Wong's Grill</rdfs:label>
   <a:foodFlavor rdf:resource="http://myresto.com#Hot"/>
   <a:foodFlavor rdf:resource="http://myresto.com#Spicy"/>
   <a:category rdf:resource="http://myresto.com#Business"/> </a:Cantonese>
```

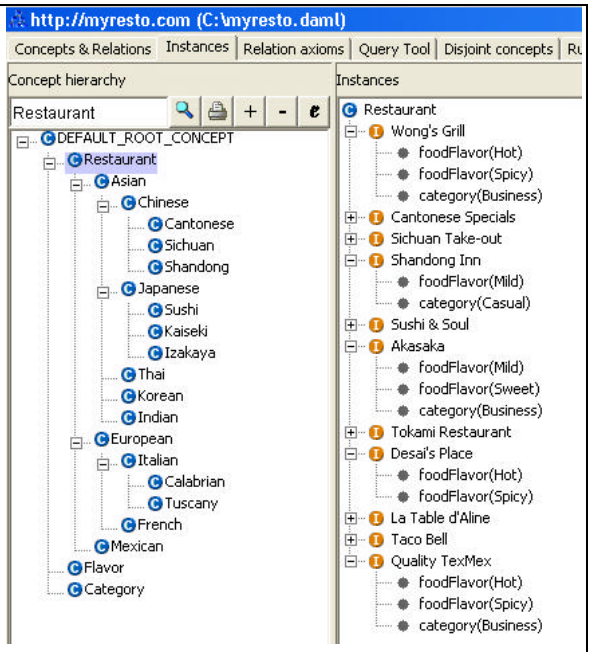

Figure 3: DAML+OIL code and respective restaurant service ontology in OntoEdit [11]

Consider our user looking for a restaurant for dinner. Specifying the request to book a restaurant with Indian cuisine at 8 p.m., the discovery could offer a booking service by a local restaurant. However, a query to book for 8 p.m. might fail, because they are already entirely booked for that time. Being cooperative a service could now offer booking at e.g. 9 p.m. and thus relax the user's constraint. The decision whether to accept the relaxed offer can only be taken by our user. But let's assume our user is for some reason constrained to 8 p.m. Then the failed selection would lead to a re-entry of the discovery with the additional information that there is no matchmaking possible for the Indian restaurant and the discovery might relax the request for Indian food towards South-Asian cuisine and discover e.g. booking services from Thai restaurants. Again querying the services will show, if booking at 8 p.m. is possible and our user can consider the new offers.

3. Semantic Service Descriptions

With the number and diversity of Web services expected to grow, enhanced techniques for service discovery and selection will be needed. An Internet wide network of Web services registries is given by the Universal Description Discovery and Integration (UDDI) [16]. It features basic information about providers of a service and the invocation details. Though UDDI has become the de facto standard in the field it also suffers from some shortcomings: UDDI is limited to keyword matching and does not support any kind of inference that would be necessary to relax descriptions along the lines of user preferences or ontologies. Therefore, even though UDDI and WSDL [5] are commonly used today to implement service catalogs, they still essentially lack strong concepts for service personalization which are crucial for advanced usability.

Research in the area of Semantic Web seeks a solution to this unsatisfying situation [3], [12]. Generally speaking, the Semantic Web fosters a population of the Web with content having formal semantics and rich service descriptions. Several semantic frameworks for Web services are currently emerging with DAML-S [1] and the Web Service Modeling Framework (WSMF) [8] as most prominent approaches. In this paper we focus on DAML-S as the most mature formal Web service framework. DAML-S is an ontology-based approach to the description of Web services trying to provide a common ontology of services for the Semantic Web. Build on top of DAML+OIL [6], the design follows the layered approach to semantic Web markup languages. The ultimate goal of DAML-S is to provide ontologies enabling agents or users to discover, compose and invoke Web services. Currently the structure of the DAML-S ontology consists of a service profile for advertising and discovering services, a process model giving a detailed description of a service's operation and a service grounding providing details on how to interoperate with services via message exchange. DAML-S also supports the classification of services by standard ontologies like DAML+OIL or OWL [4].

For advanced service provisioning in a first step we have to find out which services are executable in a sensible manner considering the given task. According to our running example, Figure 3 illustrates a semantic description of different services for restaurant booking in DAML+OIL. Here, restaurants are classified according to

their cuisine. The DAML+OIL code on the left hand side determines that e.g. 'Restaurant' is a super-class of 'Asian' or – expressed in the lingo of the Semantic Web – that 'Asian IS-A Restaurant'. In the same fashion the ontology denotes that 'Chinese' is a specialization of 'Asian' and eventually that 'Cantonese' is a specialization of 'Chinese'. Furthermore it states that all restaurant booking services are associated with the properties 'foodFlavor' and 'category' intended to describe the dominant flavors of the served food and the category restaurants belong to. For our running example we will use the catalog of semantic service descriptions shown on the right hand side of figure 3 and in figure 4. Here screenshots of the ontology workbench OntoEdit [15] that we have used to build our ontology are shown giving an overview of the service hierarchy and the available restaurant booking services. For instance the restaurant 'Wong`s Grill' is a 'Cantonese' restaurant (cf. fig. 4), serves 'Hot' and 'Spicy' food and is well suited for 'Business' arrangements (cf. fig. 3).

In some sense ontologies expressed in DAML-S give us a notion of each user's perception of relevance. [13] presents a way to map the expressive power of DAML-S to standard UDDI specifications. Hence, we can employ standard UDDI technology for our approach enriched by cooperative retrieval techniques and a suitable ontology for relaxation, if no adequate service is discovered.

4. Advertising and Cooperative Discovery

As we have seen an ontology is a partial order of terms with respect to a certain conception (generally of the 'IS-A' type). When used for the relaxation of service requests during the discovery phase we can understand this order as an instance of the 'better-than' type. Finding a service description containing a more specific term of a request is considered a better match than a description only containing a somewhat broader term. However, by offering these somewhat broader, but with respect to a certain concept related services, we will still match the user's preferences to a considerable degree and improve our provisioning.

Consider our restaurant example. Focusing on the cuisine of restaurants we use the ontology given in fig. 4. If a user e.g. looks for Cantonese cuisine and we find no adequate service, we may relax the discovery to the superclass of finding Chinese restaurants (including Cantonese, Sichuan and Shandong cuisine). Obviously, we can't decide for the user and simply book one of these restaurants, but relaxing constraints show users some possible alternatives and improves the quality of service provisioning.

SELECT grounding(service)
FROM Cantonese
WHERE booking_date = '02.10.2003' AND booking_time = '8:00 pm' AND price_range = 'medium'

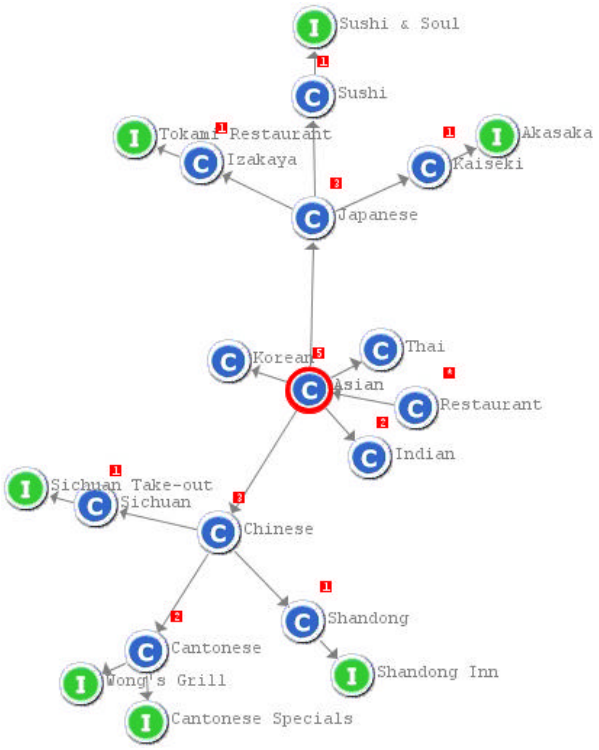

Figure 4: Restaurant services and dependencies

If no adequate services can be found, starting with the sample SQL-style declarative statement we can thus relax along the lines of our ontology ('Cantonese' is more specific than 'Chinese', 'Asian' and so on). The quality of service can further be improved, if explicit and implicit preferences of a user are considered in the discovery process. Considering for example situation-based information like a user's current location we can assume that some service requests will implicitly inherit the location:

For instance when a user in the Bay Area is looking for a restaurant for dinner we can implicitly add this information to the service request to ensure that our user won't get a recommendation for a New York restaurant. However, we must not always do so, because an expansion could prevent users e.g. preparing a trip to New York, from reserving restaurants or booking show tickets in advance.

Thus we have to know what users intend and have to understand what they assume to be common knowledge with respect to a service. In our example a user booking a restaurant would readily assume that it is common knowledge for booking a restaurant that it has to be in a certain area around his/her present location. If the user had tried to book a restaurant for a different location, it would have seemed unusual and the user would have explicitly supplied this location. In [2] we propose employing anticipated usage patterns and user profiles to model such differing user behavior. In fact these usage patterns will of-

ten even have to overwrite the general default ontologies, because the latter might interfere with user preferences and thus cause low quality cooperative behavior.

For instance a relaxation from a user's wish for Indian cuisine to the more general Asian cuisine may not be satisfying, because a specific user might have focused on the hot and spicy Indian dishes and might rather prefer a relaxation to hot and spicy Mexican food, if no service by an Indian restaurant is available.

We can thus have different conceptions of ontologies for different users incorporating both their view of the world and their personal preferences. But applying usage patterns will supply us with even more flexibility in provisioning, since personal preferences of users may change not only with the domain, but also with different intentions in the same domain and thus with different instances of otherwise similar requests for the same type of services:

When booking a restaurant a user may be driven by different intentions. There will be occasions for business dinners where users would prefer a posh French or Japanese restaurant to impress business partners or customers. But the same user might also plan for an evening out with family or friends where he/she might prefer simple Chinese cuisine. Different intentions can generally be distinguished by sets of different input parameters like the price range or restaurant ratings and/or typical values.

5. An Algorithm for Cooperative Discovery

Based on section 4 we will now state an algorithm for cooperative Web service discovery and show how it works:

Algorithm: Discovering a Service

1. Map the service request onto a best matching service category
2. Retrieve the existing usage patterns for the specific kind of service requested
3. Capture the user's intention by comparing user-provided request parameters and their values with input parameters and typical values of usage models and assign the user to a pattern
4. Decide for a suitable discovery ontology using applicable defaults or more specific user profiles
5. Discover a basic set of matching services to query for further information
6. Expand queries as shown in [2] and cooperatively retrieve offers from discovered services
7. If the user is already satisfied by one of the objects execute the respective service and exit
8. If the user is not yet satisfied, broaden the required characteristics of the service along the chosen ontology, relax the service request and proceed with step 5

Let us assume requests for restaurant booking (step 1) to show how the algorithm works. Our ontology from above focuses on the concept of regionally related cuisines which is a good guess, if nothing else is known about the user requesting a service. Let us further assume that like in our example two different usage patterns for 'casual dining' and 'business dinners' have been created (step 2). Using again a SQL-style declarative query language we can formulate the preferences of users and search for adequate services. First consider a user assigned to a 'casual' dining pattern (step 3) looking for a restaurant with Cantonese cuisine (steps 4, 5 and 6). We have to start in our ontology with the leaf 'Cantonese'. For ease of understanding we omit further attributes like date, time or price range that might have lead to assigning a pattern and which we could include in the where-clause:

```
SELECT grounding(service)
FROM Cantonese
WHERE categogy = 'Casual'
```

If this statement fails to return an adequate service or if the user is not yet satisfied (e.g. if no tables are free), we would have to relax the statement along the lines of our ontology (step 8). The next discovery would broaden the concept 'Cantonese' to 'Chinese' and repose the query keeping the category of the pattern (again steps 5 and 6):

```
SELECT grounding(service)
FROM Chinese
WHERE categogy = 'Casual'
```

This broadening would go on, until a suitable match can be found (for example the Shandong Inn). Using cooperative retrieval technology we can even put this series of statements into a single one that can be directly executed over a cooperative database [10] containing the service descriptions (using the notation of [9]):

```
SELECT grounding(service)
FROM Restaurant
WHERE categogy = 'Casual'
PREFERING (concept(service) = 'Cantonese'
    PRIOR TO concept(service) = 'Chinese'
    PRIOR TO concept(service) = 'Asian')
```

Basically this is how today's ontology matching in service provisioning works. However, we want more. We will now incorporate knowledge about the specific user for a better service provisioning. A specific user might have a very clear idea what restaurant styles he/she would like. For entertaining important customers a user might prefer expensive French or Japanese restaurants or, if that is not possible, Italian cuisine. Thus our basic ontology and a relaxation along regional broadening will no longer do. But, since our system features cooperative retrieval, we can easily incorporate such user preferences:

```
SELECT grounding(service)
FROM Restaurant
WHERE categogy = 'Business'
PREFERING ((concept(service) = 'French'
        OR concept(service) = 'Japanese')
    PRIOR TO concept(service) = 'Italian')
```

But with our flexible cooperative discovery model we can even go a step further. Let's assume a user does not state explicit preferences, but rather relies on a certain concept of cuisines. Our basic ontology gives a region-based relaxation order on the conception of geographical regions. As a default this is a very good conception because cuisines of the same region tend to be rather similar in the use of ingredients, spices, etc. For instance in our simple ontology Indian and Thai cuisine are subsumed by the same conception of both being Asian cuisines. And thus, if no adequate Indian restaurant can be found our relaxation in a first step tries to find Asian cuisine and might come up with a nice Thai restaurant.

However, a specific user might not see different cuisines related by their respective regions, but by the dominance of e.g. hot and spicy flavored food in the cuisine, i.e. have rather a flavor-based conception of the intended relaxation. Thus a request for Indian restaurants would have to be broadened with respect to the flavor concept. Given that the new conception is expressed in the ontology (here in our attribute foodFlavor) also this can be done by our cooperative discovery model having detected the different conceptions in steps 3 and 4:

```
SELECT grounding(service)
FROM Restaurant
WHERE categogy = 'Business'
   AND foodFlavor = 'Hot' AND foodFlavor = 'Spicy'
PREFERING (concept(service) = 'Indian'
    PRIOR TO concept(service) = 'Asian')
```

Given this statement and our above sample ontology the broadening would be made from Indian towards Asian services, but since there are non whose flavor is described as hot and spicy, the retrieval would due to the hard conceptual selection ignore all Asian services and finally settle on e.g. the 'Quality TexMex' restaurant. Thus our cooperative discovery respects different user conceptions.

But we still want more for advanced provisioning, because a user might have different instances of applicable concepts and preferences depending on the case of application. This is implemented by usage patterns that describe the anticipated situations in which a service request can typically be issued. For our restaurant example we have chosen two different patterns that distinguish between casually going out for dinner and having a business dinner. Assigning a user to one of the possible patterns (step 2 and 3) together with the explicitly user-provided information will thus automatically trigger the query rewriting of the respective category. If no special pattern for a user is found, the default for the pattern is adopted. The applicability of specific preferences are thus determined in each instance of requests with a pattern assigned.

User	**Pattern**	**Conception**	**Preference**
Default	casual	region-based	None
Default	business	region-based	explicit: European
Cathy	casual	flavor-based hot and spicy	explicit: Indian or Sichuan
Michael	casual	region-based	explicit: Cantonese
Michael	business	region-based	explicit: French or Japanese prior to Italian

Table 2: A set of different values for usage patterns

For instance in table 2 the default pattern for casual dining is based on geographical regions of the different cuisines and there are no specific preferences. Though also relying on the region-based conception, the business patterns states a general preference for entertaining customers in restaurants featuring European cuisines. For casual occasions Cathy has stated a conception of relaxing along 'hot and spicy' cuisines like shown above. She also would generally prefer Indian or Sichuan cuisine. Michael on the other hand has provided personal preferences for both patterns, but generally sticks to the region-based conception. Casually he prefers Cantonese cuisine, but on business occasions he'd rather take business partners to a French or Japanese restaurant and, if that fails, he would like to take them to Italian restaurants. Please note, that all settings for general patterns can be understood in the way of being common knowledge about users. Thus, if nothing else is specified in a request, they can be seen as an 'inspired guess'. However, if a user's request explicitly states a contradicting wish, it is respected.

Thus we have seen that - though using the same ontology for all users - by rewriting the cooperative discovery statement along the lines of our algorithm and posing it to a database we can effectively personalize the service provisioning. This is possible not only for explicitly provided user preferences, but also for different conceptions of certain service aspects. Moreover, suitable usage patterns will provide sensible default models with helpful settings for a variety of applications and with specific instances of applicable preferences for each user. Of course a user are enabled to overwrite patterns by explicit terms.

6. Summary and Outlook

The pervasiveness of the Internet already offers the technical possibilities to interact with helpful services anytime anywhere, but without suitable means of personalization users will not be able to make the best use of the growing number of services offered. In this paper we have focused on personalization techniques for the cooperative discovery of user-centered Web services. Unlike most business-to-business services user-centered services demand for interaction with the end user to decide, which specific service should be executed. In general this decision will depend on some additional information e.g. if a service is able to deliver a certain object or if a satisfying set of the characteristics specified in a user's request can be met. This poses a difficult problem for service providers, because especially for complex services a matching with user requests can often only be achieved cooperatively trying to negotiate a compromise between Web service and user.

We have divided the interaction between the user and the Web service into three phases: the discovery of possible services for a task, the gathering of information from these services for the selection of a special service and the subsequent execution. In this paper we focused on the cooperative discovery of suitable services using an ontology-driven approach. We have shown that during the discovery phase we can offer the user an improved quality of provisioning by successively executing service requests that are relaxed along the given service ontology and the user's personal preferences. Using the example of a restaurant booking service we have studied how we can consider not only different user preferences and various conceptions towards relaxation, but even generic usage patterns. These patterns can be used to describe varying instances of users' preferences and conceptions with respect to the typical usages of a service.

Together with the work in [2] on cooperative selection and execution of Web services our approach towards cooperative discovery promises to enable truly personalized and user-centered Web services. In [2] we have shown how techniques from cooperative retrieval, e.g. query expansion, can help to select appropriate Web services. Here, we study how cooperative retrieval and standard ontologies can be used together for the discovery of adequate services. Our future work will focus on the subsequent refinement and integration of the presented ways of personalization into standard interaction techniques. Repositories for user preferences and usage patterns as well as concepts for learning these patterns from user-service interaction will have to be investigated. Furthermore, we plan to consider specific characteristics of and heuristics for our personalization techniques when dealing with different client devices like mobile communications systems.

References

[1] A. Ankolenkar, M. Burstein, J. Hobbs, et. al. DAML-S: Web Service Description for the Semantic Web. In *Proc of the Int. Semantic Web Conf. (ISWC'02)*, Sardinia, Italy, LNCS 2342, Springer, 2002.

[2] W.-T. Balke, M. Wagner. Towards Personalized Selection of Web Services. In *Proc. of the Int. World Wide Web Conf. (WWW)*, Budapest, Hungary, 2003.

[3] T. Berner-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[4] M. Dean, D. Connolly, F. van Harmelen, et. al. OWL Web Ontology Language 1.0 Reference. http://www.w3.org/TR/owl-ref, 2002.

[5] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/2001/NOTE-wsdl-20010315, 2001.

[6] D. Connolly et al. DAML+OIL Reference Description. *W3C Note*, December 2001.

[7] Expedia Travel Portal. http://www.expedia.com

[8] D. Fensel and C. Bussler. The Web Service Modeling Framework WSMF. In *Journal of Electronic Commerce Research and Applications*. Elsevier, 2002.

[9] W. Kießling, G. Köstler. Preference SQL - Design, Implementation, Experiences. In *Proc. of the Int. Conf. on Very Large Data Bases (VLDB'02)*, Hong Kong, China, 2002.

[10] J. Minker. An Overview of Cooperative Answering in Data-bases. In *Proc. of the Int. Conf. on Flexible Query Answering Systems (FQAS)*, Roskilde, Denmark, LNCS 1495, Springer, 1998.

[11] Ontoprise. Ontology Engineering Workbench – OntoEdit. http://www.ontoprise.com

[12] M. Paolucci, T. Kawamura, T. Payne, K. Sycara. Semantic Matching of Web Services Capabilities. In *Proc. of the Int. Semantic Web Conf. (ISWC'02)*, Sardinia, Italy, 2002.

[13] M. Paolucci, T. Kawamura, T. Payne, K. Sycara. Importing the Semantic Web in UDDI. In *Proc. of the Int. Workshop on Web Services, e-Business and the Semantic Web (WES'02)*, Toronto, Canada, 2002

[14] SOAP Protocol. http://www.w3.org/2000/xp/Group.

[15] Y. Sure, M. Erdmann, J. Angele, et. al. OntoEdit: Collaborative Ontology Development for the Semantic Web. In *Proc of the Int. Semantic Web Conf. (ISWC)*, Sardinia, Italy, LNCS 2342, Springer, 2002.

[16] UDDI. The UDDI Technical White Paper. http://www.uddi.org.

[17] M. Wagner, W.-T. Balke, R. Hirschfeld, W. Kellerer. A Roadmap to Advanced Personalization of Mobile Services. In *Proc. of the Int. Conf. DOA/ODBASE/ CoopIS (Industry Program), Irvine*, CA, 2002