PERSONALIZED WEB SERVICE DISCOVERY

Sanaullah Nazir, Brahmananda Sapkota and Tomas Vitvar
Digital Enterprise Research Institute, National University of Ireland, Galway, IDA Business Park, Galway, Ireland

Keywords: Semantic Web, Semantic Web Services, Ontology, Semantic Web service discovery, e-Government, Personalization.

Abstract: Most of the existing work in web service discovery reflect providers perspective. These discovery techniques improve service descriptions so that the service clearly describes what it offers. This development has led to the use of semantic web technology for web service discovery. A lot of work has gone into improving web service descriptions to support improved web service discovery. In this paper we extend the existing approaches to semantic web service discovery and introduce personalization to it. We also show how personalization affects discovery. We provide a model to represent a user profile, and its usage for improving the accuracy of web service discovery.

1 INTRODUCTION

Advancement in web service discovery is mostly centred around the service provider. Use of UDDI (Clement et al., 2004) has been amongst the initial efforts for improving web service discovery. The inability of web service description language WSDL to describe what the service provides led to the use of semantic technology, involving DAML-S (Paolucci et al., 2002), OWL-S (Srinivasan et al., 2004), WSMO (Keller et al., 2004) etc. Most of the currently used discovery approaches has been geared towards identifying services which meet the user's requirements. There might be some web services, which provide the same output, but the user of the service could be different.

Context based service discovery (Hai et al., 2005), (Maamar et al., 2006) is a type of service discovery approaches which concentrate on the user's perspective. But the direction of context based discovery has been dependent on environmental parameters in which the user is searching for a service.

Another discovery approach is QoS based service discovery (Vu et al., 2005), (Makripoulias et al., 2006). In this approach some numeric parameters are set for some characteristics of a service. These values should match with the requirements of the service user, e.g the required shipment time of the product should match with the shipment time provided by the service.

Here it is important to understand that the discovery approaches mostly try to consider what the user requires without considering the profile of the user. In this paper we present a discovery approach involving the user profile as a functional requirement of the service to be matched as well. Answers to the questions such as how a goal can evolve in order to make discovery more accurate, and how the underlying storage architecture supports discovery.

The structure of this paper is as follows: Section 2 provides the motivating example, which we use to present our approach of service discovery. Section 3 outlines modeling of web services and goals and Section 4 presents the how the discovery of web services takes place and how goal evolves during the process. This section also describe the peer-to-peer architecture on which the discovery component is relying on. Section 5 describes the tools used to implement the whole architecture, a run through example for the case study is elaborated in section 6. Section 7 presents some of related work done in this direction. Finally, Section 8 draws some conclusions and presents future directions.

2 MOTIVATING EXAMPLE

In this section we will present a motivating scenario for this research work. In this example different variety of driving licences are issued depending on the

Nazir S., Sapkota B. and Vitvar T. (2008).
PERSONALIZED WEB SERVICE DISCOVERY.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 82-89
DOI: 10.5220/0001526400820089

person's physical ability and the type of vehicle he drives. The input required from the user is a type of vehicle, whether he holds international/EU-State driving licence, and if he has any disability. The personal profile input would allow the proper discovery of required service(s). If in this case user profile isn't considered then a large amount of services will be discovered from which the selection would have to be done by the user.

3 SERVICE AND GOAL MODELLING

In this section we present language and technique used to model the web services and user goals. For describing web services and user goals WSML-Flight variant of WSML (Toma, 2007) is used. WSML-Flight allows writing simple logical expressions and rules. Web services, goals and ontologies are modeled using WSMO (Roman et al., 2007).

The reason for modeling and using personalization is to provide different discovery support from user to user, depending on his needs, status and environment, rather than providing crude services. Some work towards personalization for information retrieval has been done in digital libraries (Amato and Straccia, 1999), where users are modeled in terms of five categories, which are personal data, preferences and restrictions, time and location of delivery, high responsiveness of the service to user's need and privacy of the user data. Moving towards the web, (Garrigos et al., 2005), Garrigos has presented reusable models for personalization. In their approach five models are presented. Domain model to defined the structure of domain data, Navigation model to define structure and behavior over the domain data. Presentation model is used to handle the layout of the presentation. The personalization model holds rules for different personalization policies and User model allows to store data needed for the personalization.

In this paper we model the user in terms of his needs and profile. QoS parameters i.e price, delivery time etc are not handled in our model, but are considered to be a non-functional property of the service which the user can negotiate later.

We have modeled the need and profile of the user as a functional property of the service and the goal. Using WSMO for modeling service and user goal, both the profile and the need will be in the funcational description. The profile of the user is defined as a concept in a domain ontology. The reason for having this concept in domain ontology is the varying requirement of profile for the domains. User of the service can be an individual or an organization. A service may be a user of another service, but the profile used will be of the external entity. Some profile inputs may be required for one domain, while those might not be of interest in the other. The service is also dependent on the location of the user, e.g the driving licence service is provided by many countries, but the location input provided by the user will allow providing a more accurate discovery.

A web service W can be modeled in WSMO as follows:

$$\mathrm{W} = \{\mathrm{F}, \mathrm{B}\}$$
$$\mathrm{F} = \{W\rho, W\rho'\}$$
$$\mathrm{B} = \{\sigma, \sigma'\}$$

F represents functional description of the web service and B represents the behavioral description of the web service. A functional description of web service contains precondition $\mathrm{W}\rho$ and postcondition $\mathrm{W}\rho'$ of a web service. WSMO provides two more functional descriptions attributes which are assumption and effect. But we are only using precondition and postcondition.

The behavioral description of the service B contains choreography σ and orchestration σ'. The choreography of a web service describes the communication pattern and how the service can be used. Orchestration represents how the overall functionality can be achieved by cooperation of different web service.

The precondition of web service contains rules which allow to verify the user's profile. The profile concept created in the domain ontology is used in the precondition to evaluate the state of the information before service execution. The precondition can be defined as follows:

$$W\rho = \alpha$$

Where

$$\alpha = \{\nu 1, \nu 2, \nu 3, \cdots, \nu n\}$$

Here α is the set of conditions. This set contains individual conditions νi which needs to be fulfilled before the execution of the service. A service can have multiple preconditions; all of them are need to be fulfilled by the user goal. Starting from very generic precondition and moving towards specific precondition for the services. The generic precondition is held by all the services that provide same output. Moving from generic preconditions towards specialized preconditions allows differentiating the profile input required by the various service of the same type.

Service type is basically what a service provides. Web services are said to belong to the same service type if they have a common postcondition, but a varying precondition.

This will allow to repeatedly evolving a goal which can accurately match with a web services or a set of web services.

The postcondition of the web service description defines the state of the information that is reached after the service is executed. The need of the user is modeled in the postcondition of the service. There can be web service which have common postcondition but the precondition they require to be verified are different. Postcondition can be defined as follows:

$$W\rho' = \{\beta\}$$

Here β represents the state of the information after execution of the service.

Benefit of using WSMO for modeling is that it provides support for modeling the user's request. The user's request is divided into two parts, as need of the user and profile of the user.

We follow a similar approach for modeling goal as presented in (Stollberg et al., 2007). We use goal template and goal instance. Goal template can be defined as follows:

$$G' = \{G\rho, G\rho'\}$$

A goal template G' contains precondition $G\rho$ and postcondition $G\rho'$. Precondition of a goal template is the generic precondition which all the services in one service types define.

Following is how goal is defined, Let G represent the WSMO goal:

$$G = \{G\rho'\}$$

Here $G\rho'$ is the postcondition of the goal. User's need is captured by the postcondition of the goal. We are not using precondition in the functional part of the goal. The reason of not using precondition is difficulty for the user to write logical expressions. The user is not interested defining the conditions on the information but rather would like to provide his input as simple data. Thus for supporting the user to define his profile, we use the user's profile concept described in the domain ontology.

The user request then finally comprises of WSMO goal containing postcondition and user data as user's profile concept.

Goal templates are provided by service provider and are created at design time. Transition from goal template to goal takes will be explained in the goal creation section presented later. But for now, goal template is used as a starting point for creating a goal instance.

4 ARCHITECTURE OVERVIEW

4.1 Goal Creation and Service Discovery

This section elaborates the user's perspective of the overall architecture and how a user's request is formalized which is later used for discovery. WSMO provides support for modeling a goal, but it does not provide the technique for creating such a goal. An effort made towards this technique is presented in (Stollberg and Norton, 2007) . In this paper Stollberg has presented two forms of goals namely Goal templates and Goal Instance. He makes further classification in Goal templates, which are basic type and composed type. Our work is some what along the same lines. Here we currently only consider goals of basic type. Goal will be modeled using the goal modeling technique presented in the previous section. The goal evolves for an accurate discovery of service. As defined in the previous section a goal template contains precondition from the generic type which belong to all the web service of the same type. A user requests for a web service goal template using key words. The matching process retrieves the goal templates which match his request. User input is required for all the precondition variables. The precondition is used to create a user profile which is sent with the goal in a form of instance of the concept. Evolution of goal takes place during the web service discovery process. Each partial input is matched with a complete description of web service. Incomplete inputs become a part of precondition and are requested in the next step. The process can be represented as follows: Let initial partial goal created from goal template be

$$GoalTemplate = G'\{G\rho, G\rho'\}$$
$$G\rho = \alpha\{\nu 1/t1\}$$

Here ν's are free variables. These variables are replaced by the t input given by the user.

$$\text{User Data } \{\delta\} \subset \alpha\{\nu 1/t1\}$$

This precondition results into creation of user data as the concept of a profile in an ontology. This concept satisfies the precondition of the goal template.

From a goal template a goal is created having only the postcondition of the goal template. Input for discovery is $\rho', O(\delta)$ goal with postcondition and ontology containing user data profile concept. Through the discovery process user profile is refined. The precondition not matching is returned from the discovery component. Just as the case with goal template this precondition is used to take input from user to extend the user profile. Next the precondition with this extended data is sent again to discovery component.

This process continuous until all the precondition of the service is satisfied. The following relation must hold:

$$G\rho \subset W\rho$$

Since

$$G\rho \subset \alpha\{v1/t1, ..., vn/tn\}$$

And

$$\delta \subset \alpha$$

Then

$$\delta \subset W\rho$$

This relation explains that the user data should hold for the precondition of the web service. The postcondition of the goal must also hold for the web service postcondition. Following is the relation for postcondition.

$$G\rho' \subset W\rho'$$

The process of discovery first matches postconditions. Web services which have matching postconditions are then taken for matching the preconditions of the service. The output from the discovery component is list of web service along with precondition which need to be satisfied. If the web service has no more preconditions to be matched then the second attribute is null. From the list the user selects which web service fulfils his need. On selecting a service with un-fulfilled precondition, then the user has to provide input.

Here the deviation from the traditional approach is taken, where the following relation must hold:

$$G\rho \subset W\rho$$
$$G\rho' \subset W\rho'$$

Creating an instance of a concept from the input provided by the user is simpler. This would save the user from writing complex logical expressions and for component to manipulate with these expressions, without complete knowledge of the domain. This instance data will allow creating simple queries from the web service description to verify if the instance holds.

4.2 Storage Requirement

In this section we present the storage architecture of our discovery approach. The underlying storage architecture is based on peer to peer architecture as presented in (Sapkota et al., 2007). Service providers / agency providing the web service description acts as a peer and becomes a part of the semSet which is of the same domain. There are few changes made in the overlay network, to cope with the requirement of the e-gov domain. The domains of the service are clearly defined. The peer can belong to multiple domains. Another e-gov domain specific restriction is the privacy of the member state i.e to keep data of one member state unaccessible to the other member state. There is a communal gateway which provides mediation between ontology. Semantic query (WSMO goal) requiring interaction with other member state passes through the communal gateway. More detail regarding the communal gateway can be found in (Vitvar et al., 2006). The mediation approach used can be found in (Mocan, 2005), which is a part of WSMX architecture (Zaremba and Moran, 2005). WSMX (*web service execution environment*) is running on all nodes. The discovery and storage components are part of WSMX architecture. Each member state has its own coSet. From the member state coSet one of them becomes the member of inter-member state level coSet. Inter-state querying takes place through these peers. For details about the construction and membership algorithm of the network refer to (Sapkota et al., 2007). In figure. the doted line ring represent the semSet. Where there is only one peer in the semSet, that peer is a member of the member state coSet. The dashed black line represents the member state coSet.

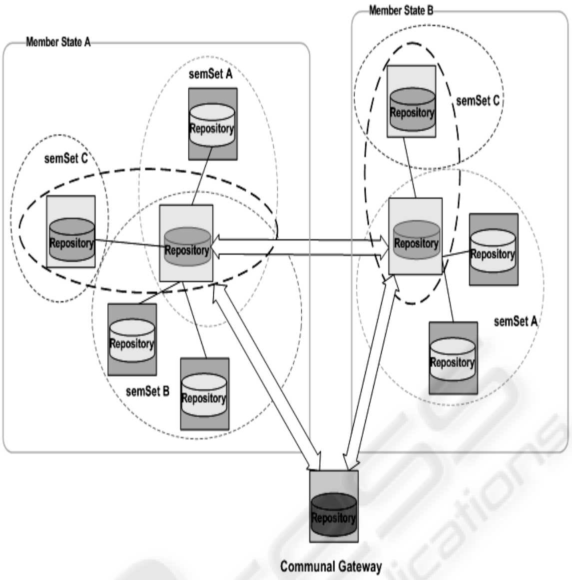

Figure 1: Peer-to-Peer Architecture for discovery.

5 IMPLEMENTATION

The discovery component provides the functionality of discovery of semantic web services which are most suitable for the user. Figure 2, shows structure of each peer. Discovery component performs matchmaking between WSMO goal and profile provided as an instance of citizen concept with the semantic service descriptions. Personalization component handles the ordering of personalization rules in the preconditions needed to be verified by the discovery component, and returns the formatted output to the external component. The Repository holds the semantic data and communication engine handles the peer-to-peer communication.

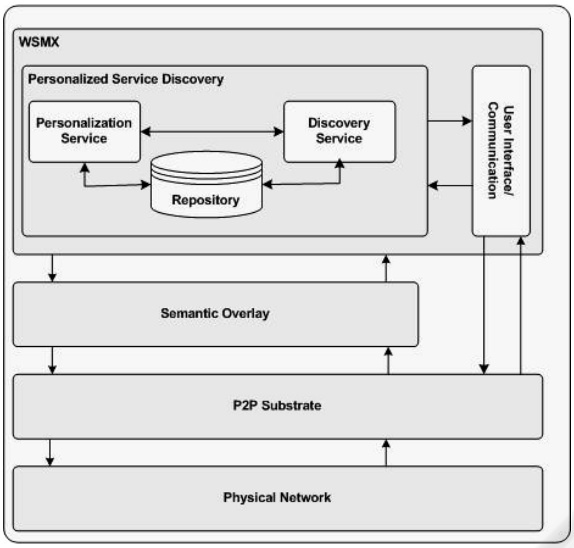

Figure 2: Peer Structure.

The implementation has been done in Java 1.5. We have used the WSML2Reasoner framework (Motik et al., 2005), and KOAN2 reasoner for evaluation of logical expressions defined in the semantic descriptions of the services. The underlying communication between the peers is handled by JXTA (Li, 2003). The semantic repository used is ORDI (Ognyanov and Kiryakov, 2005).

6 RUN THROUGH

In this section we present a running example for the case study. The requirement is that the user needs a Irish driving licence. In this example we will consider that the ontology being used is same for goal and web service. Non-Functional property(nfp) includes the domain and member state the entity belongs to. These nfps are used for routing purpose between the peers. Table 1 and Table 2 shows how the web service, goal template and goal are described.

Figure 3, shows how the goal evolves and discovery takes place. Initially the user is provided with a set of keyword, which allows him to select what his need is. Using these keywords goal templates are matched. From the list of goal templates the user selects one.

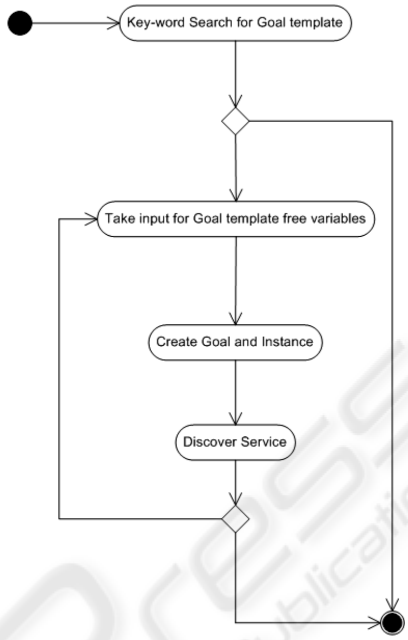

Figure 3: Discovery Process.

The goal template as defined in the modeling section has precondition with free variables. Input is taken from the user for these free variables. These input result in the creation of instance of the person concept defined in the ontology. This instance along with the goal having the postcondition only is sent to the discovery component. The discovery component performs matchmaking and as a result returns a web service / set of web services, along with free variable which require more user input. If the service has no corresponding free variable it is considered to be discovered. As the user selects a goal template from the list of goal template, here again the user has to select a web service from the list of web services, which serves in the same way as goal template. This activity continuous until no more free variable are available along with the service, or the user is satisfied with the discovered service.

In this case study, there are services which provide different type of driving licences. Here just for simplicity and to present our point we consider only one type of licence. The concentration in this example is to present how the goal evolves and how precondition of the web service plays its part in the discovery processes. The keyword based approach to select the goal template is used. This goal template asks for user to fill the nationality, age and if the user has any physical disability. This resulting in the creation of a goal without precondition and instance data. Next this goal is sent for discovery. The discovery component returns ”Web service A” that matches the goal and instance

Table 1: Web Services.

webService _"webServiceA.wsml" $W\rho = \alpha\{\nu 1, \nu 2\}$, $W\rho' = \{\beta\}$ $\nu 1$ = ?req{Nationality ="IrishCitizenship" $\wedge$ age $\geq$ 18 $\wedge$ disable = "false"} memberOf request. $\nu 2$ = ?req{IrishId = "true"} memberOf request. β = ?res{VehicleType = "car" } memberOf drivingLicence.	This web service provides a driving licence of Ireland for a car. The generic precondition for this service is that the person is an Irish citizen and has no disability. For specialized precondition the citizen holds an Irish id as well.
webService _"webServiceB.wsml" $W\rho = \alpha\{\nu 1, \nu 2\}$, $W\rho' = \{\beta\}$ $\nu 1$ = ?req{Nationality ="IrishCitizenship" $\wedge$ age $\geq$ 18 $\wedge$ disable = "true"} memberOf request. $\nu 2$ = ?req{IrishId = "true"} memberOf request. β = ?res{VehicleType = "car" } memberOf drivingLicence.	This web service provides a driving licence of Ireland for a car. The generic precondition for this service is that the person is an Irish citizen and has some disability. For specialized precondition the citizen holds an Irish id as well.
webService _"webServiceC.wsml" $W\rho = \alpha\{\nu 1, \nu 2\}$, $W\rho' = \{\beta\}$ $\nu 1$ = ?req{Nationality ="EU" $\wedge$ age $\geq$ 18 $\wedge$ disable = "false"} memberOf request. $\nu 2$ = ?req{EUId = "true"} memberOf request. β = ?res{VehicleType = "car" } memberOf drivingLicence.	This web service provides a driving licence of Ireland for a car. The generic precondition for this service is that the person is an EU citizen and has no disability. For specialized precondition the citizen holds and EU member state id as well.

Table 2: Goal evolution.

goal _" goalTemplate.wsml" $G\rho = \alpha\{\nu 1\}$, $G\rho' = \{\beta\}$ $\nu 1$ = ?req { Nationality = ?nat $\wedge$ age = ?age $\wedge$ disable = ?dis} memberOf request. β = ?res{VehicleType = "car" } memberOf drivingLicence.	This goal template provides a starting point to discover a service which provides a deriving licence of Ireland for a car. The precondition provide what information is required from the user. The free variables are used to taking input. The postcondition defines that the user is looking for a web service which provides car driving licence. This goal template is created at design time by the service provider.
goal _" goalInit.wsml" $G\rho'$ ={β}, O = {δ}, β = ?res{VehicleType = "car"} memberOf drivingLicence. δ = ?data{Nationality ="IrishCitizenship" $\wedge$ disable = "false" $\wedge$ age = 20 } memberOf Citizen.	This goal is created after the user's input for the free variables. The instance created has all the data which is needed for discovery
goal _" goalFinal.wsml" $G\rho'$ ={β}, O = {δ}, β = ?res{VehicleType = "car"} memberOf drivingLicence. δ = ?data{Nationality ="IrishCitizenship" $\wedge$ age = 20 $\wedge$ disable = "false" $\wedge$ IrishId = "true"} memberOf Citizen.	This is the second level of the goal, where after the first iteration web service discovered, i.e. "web service A" requires further input for precondition. To check if the person holds an Irish Id card.

data, which is if the user has the Irish Id. This precondition like the goal template is used to take input from the user and to update the instance data. This final goal and instance data is sent for discovery which finally only returns "Web service A" with null for further input requirement. The routing of the goal takes place using the nfps. Once the peer receives a request for discovery, it uses the nfps to check to which peers to route the query to. Peer receiving the request has in its routing table peers who belong to that domain and member state, it forwards the goal to those peers. In case the peer holds no information for that domain or member state, it forwards the request to the coordinator peer. This forwarding of request continues until the inter-member state coordinator is reached. The coordinator forwards the request to the member state to which the query belongs to. The result is returned along the same path taken to reach the node.

7 RELATED WORK

There has been a lot of work done for web service discovery. Most of the work concentrates on the simple web service discovery without considering the context or the personalized view of the user.

The two phase discovery presented in (Stollberg et al., 2007), which we extend to create goal for discovery, provides a way to model the goal for discovery and discover web service, but does not take into consideration about the possible use of precondition for providing accurate web services.

Wolf-Tilo presents a personalized selection of web services approach in (Balke and Wagner, 2003). The methodology presented there handles the task of personalization at selection time. The discovery approach used in keyword based. Ther discovery proposal lacks expressiveness of the query and the service. They use simple SQL query to select the services that meet the user's requirement.

(Maamar et al., 2006), gives a very comprehensive view for modeling the personalization aspect for web service discovery. They present a discovery methodology which requires interaction with the web services and the resource where the web service has to be executed. They consider three aspects when considering about the personalization, which are User context, web service context and resource context. User context considers user's preferences, including where and when the service is needed to be executed. The web service context considers the constraints on the web service to execute in a specific environment and lastly the resource context is the time and load constraint on the resource. Their approach is quite similar to our in terms user's view, but not much detail is presented regarding the modeling language of the service and query from the user.

8 CONCLUSION AND FUTURE WORK

In this paper an approach for discovering web services has been presented. This papers considers discovery from users perspective. Creation of refined goal from a generic one is provided to make an accurate service discovery. Modelling of web services and goal has been provided using WSMO. WSMO allows to model a profile of the user as ontology concept. We have presented how the use of precondition of the WSMO web service description can be done, which earlier has not been used. A distributed architecture has been presented, which handles distributed querying. This has resulted in efficient discovery of services as the load on the discovery component has been distributed. The inferencing time required for discovery is reduced.

This architecture has been implemented in the EU funded SemanticGov, and has result in very accurate service discovery during its testing. More work needs to be done for handling composed goals. Creating sub-goals and handle discovery and aggregating results. This has been left for the future work.

ACKNOWLEDGEMENTS

This material is based upon works supported by the EU funding under the projects TripCom (FP6-IST-4-027324-STP), SemanticGov (FP6-IST-4-027517) and by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131.

REFERENCES

Amato, G. and Straccia, U. (1999). User profile modeling and applications to digital libraries. In *Proc. of 3rd European Conference on Research and Advanced Technology for Digital Libraries*, pages 670–683.

Balke, W. and Wagner, M. (2003). Towards personalized selection of web services. In *Proc. of 12th World Wide Web Conference*.

Clement, L., Hately, A., Riegen, C., and Rogers, T. (2004). Uddi version (3.0.2), uddi spec technical committee draft. Technical report. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

Garrigos, I., Gomez, J., Barna, P., and Houben, G. (2005). A reusable personalization model in web application design. In *Proc. of Workshop on Web Information Systems Modelling*, pages 40–49.

Hai, L., Qing, L., and Naijie, G. (2005). Quantifying contexts for user-centered web service discovery. In *Proc. of 10th IEEE International Enterprise Distributed Object Computing Conference*, pages 399–404.

Keller, U., Lara, R., Polleres, A., Toma, I., Kifer, M., and Fensel, D. (2004). *D5.1 v0.1 WSMO Web Service Discovery WSML Working Draft*.

Li, S. (2003). Jxta 2: A high-performance, massively scalable p2p network. Technical report.

Maamar, Z., Mostefaoui, S., and Mahmoud, Q. (2006). Context for personalized web services. In *Proc. of 38th Hawaii International Conference on System Sciences*.

Makripoulias, Y., Makris, C., Panagis, Y., Sakkopoulos, E., Adamopouloua, P., and Tsakalidis, A. (2006). Web service discovery based on quality of service. In *Proc. of 4th ACS/IEEE International Conference on Computer Systems and Applications*, pages 196–199.

Mocan, A., editor (2005). *WSMX Data Mediation*. WSMO Working Group.

Motik, B., Nagypal, G., and Grimm, S. (2005). Wsml reasoner. Technical report.

Ognyanov, D. and Kiryakov, A. (2005). Ontology representation and data integration (ordi) framework. Technical report.

Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. (2002). Semantic matching of web service capabilities. In *Proc. of 1st Int. Semantic Web Conference*, pages 333–347.

Roman, D., Lausen, H., and Keller, U., editors (2007). *The Web Service Modeling Ontology (WSMO)*. WSMO Working Group.

Sapkota, B., Nazir, S., Vitvar, T., Toma, I., Vasiliu, L., and Hauswirth, M. (2007). Semantic overlay for scalable service discovery. In *Proc. of 3rd International Conference on Collaborative Computing: Networking, Applications and Worksharing*.

Srinivasan, N., Paolucci, M., and Sycara, K. (2004). Adding owl-s to uddi, implementation and throughput. In *Proc. of 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC)*.

Stollberg, M., Keller, U., Lausen, H., and Heymans, S. (2007). Two-phase web service discovery based on rich functional descriptions. In *Proc. of 4th European Semantic Web Conference*, pages 99–113.

Stollberg, M. and Norton, B. (2007). A refined goal model for semantic web services. In *Proc. of 2nd International Conference on Internet and Web Applications and Services*, pages 17–22.

Toma, I., editor (2007). *The Web Service Modeling Language (WSML)*. WSMO Working Group.

Vitvar, T., Kerrigan, M., Overeem, A., Peristeras, V., and Tarabanis, K. (2006). Infrastructure for the semantic pan-european e-government services. In *Proc. of AAAI Spring Symposium on The Semantic Web meets eGovernment (SWEG)*.

Vu, L., Hauswirth, M., and Aberer, K. (2005). Towards p2p-based semantic web service discovery with qos support. In *Proc. of 3rd Business Process Management Workshops*, pages 18–31.

Zaremba, M. and Moran, M., editors (2005). *WSMX Architecture*. WSMO Working Group.