A Peer to Peer Dynamic Framework for Seamless Publication and Discovery of Semantic Web Services

Thomi Pilioura*

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
thomi@di.uoa.gr

Abstract. The challenge of publishing and discovering web services has recently received lots of attention. Various solutions to this problem have been proposed which, apart from their offered advantages, suffer from the following disadvantages: (i) most of them are syntactic-based leading to poor precision and recall, (ii) they are not scalable to large numbers of services and (iii) they are incompatible, thus yielding in cumbersome service publication and discovery. This paper presents the principles, the functionality and the design of PYRAMID-S which addresses these disadvantages by providing a scalable framework for unified publication and discovery of semantically enhanced services over heterogeneous Registries. PYRAMID-S uses a hybrid peer-to-peer topology to organize web service Registries based on domains. In such a topology, each Registry retains its autonomy, meaning that it can use the publication and discovery mechanisms as well as the ontology of its choice.

Keywords: web service publication, web service discovery, framework, unified, scalable, semantic web services, PYRAMID-S.

1 Introduction

Web services (abbr. WS) have emerged as a dominant set of recommendations and standards. They have marked current web engineering methodologies and are ubiquitously supported by IT vendors and academia. The rise on the WS consumption brought forward the problem of locating the most appropriate services to use from the vast number of available ones [16]. Different solutions to this problem have been proposed (e.g. UDDI [10], ebXML Registry/Repository [8][9], SPiDeR [13], MWSDI [17], DIRE [1], OWL-S/UDDI Matchmaker [12] and WSMX [5]), each with its specific model and realization. However, the effectiveness of these solutions is limited due to the following reasons:

- A large number of these solutions are syntactic-based. This means that only syntactic information is used in the service advertisement, in the service query and in the matchmaking process. Syntactic information describes the *interface of services* and *how* and *by whom the services are deployed*. Thus, it provides informa-

* Dissertation Advisor: Aphrodite Tsalgatidou, Assistant Professor

tion about the rules, structures and terms that someone should use in order to communicate with a service. However, the use of only syntactic information in the service advertisements, service queries and in the matchmaking process leads to discovery results of poor quality.

- Most of these solutions are not scalable meaning that they are not able to scale to large numbers of services, service publishers and service requesters. This is due to the fact that they mostly follow a centralised Registry approach. The scalability issue of centralised approaches is usually addressed with the help of replication (e.g. UDDI). However, replicated Registries have high operational and maintenance cost. Furthermore, they are not transparent due to the fact that updates occur only periodically.
- Current solutions are incompatible, a fact that further aggravates the above situation. Consider for example the case of an international company the subsidiaries of which provide lots of web services stored in Registries of heterogeneous types (e.g. UDDI, ebXML Registry/Repository, SPiDeR or MWSDI). If someone wants to publish a service in these Registries, he or she has to understand the mechanisms supported by each Registry type and then separately employ these mechanisms in order to publish the service. Similarly, in case of service discovery the user has to invest considerable time visiting numerous Registries of the same or different type, understanding the way to use them, entering search criteria repeatedly and integrating potentially heterogeneous replies. We argue that the WS publication and discovery process could be greatly facilitated if the user entered publication data or search criteria only once, i.e. if the publication or discovery process took place uniformly over the various heterogeneous service publication and discovery mechanisms.

This research focuses on addressing the aforementioned situation. In particular, a framework for web service publication and discovery, called *PYRAMID-S*, is proposed. This framework has the following main contributions:

- Unified web service publication and discovery
  - over heterogeneous Registries, thus alleviating the users from the burden of handling the diversion between different technologies
  - based on syntactic, semantic and Quality of Service (QoS) information improving in this way the precision and the recall
- Preservation of the autonomy of web service Registries by allowing the accommodation of different publication and discovery mechanisms
- Use of a scalable infrastructure which organizes Registries based on domains

The paper is organized as follows. In Section 2, a description of the PYRAMID-S principles is provided. Following, in section 3, the PYRAMID-S functionality and design is presented. The paper ends, in section 4, with the main points of the research and its contribution in the specific research area.

2 PYRAMID-S Principles

Towards the confrontation of the disadvantages of current approaches in web service publication and discovery, the PYRAMID-S framework has been proposed. This

framework is based on several principles, namely a layered architecture, a number of ontologies, a service description language and a service query language as well as the concept of mediation. All these principles are presented below.

The PYRAMID-S framework addresses the problem of *scalability* by categorizing Registries to domains (e.g. Banking, Healthcare, Tourism) and by using a layered architecture. Registry categorization helps in filtering the registries where the web service publication or discovery will take place thus improving the performance. The layered architecture consists of three layers depicted in the right part of Fig. 1 and presented below:

- The *Registries Layer* consists of a number of *Registries* provided by diverse Registry operators. The Registries are responsible for getting the service advertisements or the service queries and for performing the necessary actions.
- The *Gateways Layer* consists of a number of servers, called *Gateways*, which function as entry points to the PYRAMID-S system and provide to the clients the ability for unified publication and discovery. Furthermore, they provide the appropriate interface for the management of ontology and registry information.
- The *Routers Layer* consists of a number of servers, called *Routers*, which hold ontology and registry information, including the categorization information of registries to domains. The last is used by the Routers in order to provide routing service to the Gateways in order to forward the queries or the advertisements entered in them to the appropriate domain registries.

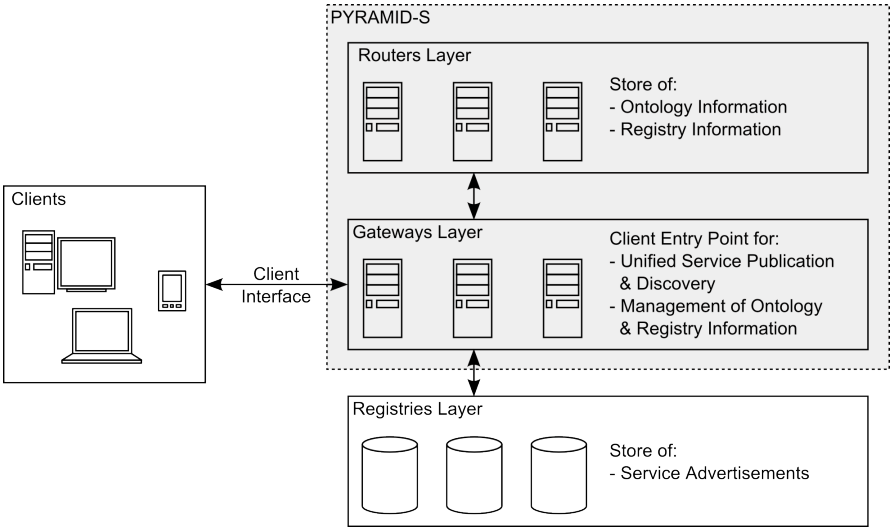

Fig. 1. The PYRAMID-S Conceptual Architecture

The layered architecture is based on a peer-to-peer infrastructure designed and implemented in the scope of this research. The peer-to-peer infrastructure is primarily designed to facilitate peers in locating one another and in "working together" in groups and more precisely in performing commonly approved actions. The last is achieved through the use of a *Peer Group Decision Making Mechanism* which has been designed and implemented in the scope of this research. The above infrastructure

is generic and it can be used in any application that requires the aforementioned functionality. This peer-to-peer topology renders PYRAMID-S scalable as it allows Routers and Gateways to easily join and leave the peer-to-peer network. More importantly, this topology ensures that there is no single point of failure in the Routers and Gateways layer. Overall system performance and scalability may be tuned by adapting the number of Routers and Gateways and according to business requirements.

The problem of *low precision and recall* was addressed by enriching the service advertisements and the service queries with semantic and QoS information. This is achieved with the help of ontologies and two special languages. The ontologies proposed in this research are the *Standard Domain Ontology (SDO)*, the *Registry Domain Ontology (RDO)* and the *Domain Classification Ontology (DCO)*.

- The Standard Domain Ontology (SDO) reflects abstract concepts and relationships in a particular application domain. It has two parts: the *Operation* part and the *Data* part. The Operation part models major action types and thus helps to determine the type of operations that each web service performs. The Data part incorporates concepts, their properties and relations among concepts in a particular application domain. The SDO is constructed by domain experts and it is mandatory to associate an SDO to each domain in PYRAMID-S. The SDO of a specific domain is the default ontology of the PYRAMID-S framework for that domain. The Registries conforming to that SDO use this ontology for the semantic publication and querying of the web services they hold.
- The Registry Domain Ontology (RDO) is defined in case of Registries that do not conform to the SDO. In this case, the Registry operator has to provide a mapping [2][3][4] from its own ontology to the SDO.
- The Domain Classification Ontology (DCO) holds information about the relationships among domains, the mappings of each registry participating in PYRAMID-S to one or more domains, the relationships between domains and SDOs and the properties of Registries, such as the access URL, the Registry type, the Registry provider details, the access URL of the RDO to SDO mapping (in case of non-conformance to SDO) and the constraints in accessing that Registry.

PYRAMID-S uses the PS-WSDL language, which is an extension to WSDL [18] proposed in this research, for describing web services and the USQL for the formulation of service queries.

PS-WSDL [11] provides the following additional information about a service: service capabilities, geographic scope, domain, QoS attributes of the service as well as information about the service provider. Fig. 2 depicts part of the PS-WSDL advertisement of a CreditScoreCalculation service. A test case ontology for the domain of Loan Services is used in order to annotate the operation, input and output elements. The NAICS taxonomy [7] is used to specify the service domain. The QoSMetrics element is used for specifying QoS properties of the service by using the WS-QoS specification [14]. Elements of the Geographic Classification System (ISO 3166-1999) [6] are used to depict the geographic scope of the service. Finally, the businessEntity element contains information about the service provider.

```
<interface name="CreditScoreCalculationIF">
  <operation name="getCreditScore" patern= http://www.w3.org/2004/08/wsdl/in-out
     concept="http://www.nbg.gr/LoansServicesSDO.owl#Credit_Score_Calculation">
    <input messageLabel="In" element="CreditScoreCalculationRequest"
     concept="http://www.nbg.gr/LoansServicesSDO.owl#Customer_SSN" />
    <output messageLabel="Out" element="CreditScoreCalculationResponse"
     concept="http://www.nbg.gr/LoansServicesSDO.owl#Credit_Score" />
  </operation>
  <domain domainName="Commercial Banking" taxonomyURI="http://ww.naics.com/"
     domainCode="522110" />
</interface>
<service name="CreditScoreCalculation" interface="tns:CreditScoreCalculationIF">
  <endpoint name="CreditScoreCalcEndPoint"
     binding="tns:CreditScoreCalculationIFBinding"
     address="http://www.di.uoa.gr/~thomi/pyramid-s/CreditScoreCalculation">
    <QoSMetrics xmlns="http://www.wsqos.net/schemas/">
      <definition>
        <offers>
          <include url=" http://www.di.uoa.gr/~thomi/pyramid-s/QoSOffer.wsqos"/>
        </offers>
      </definition>
    </QoSMetrics>
  </endpoint>
  <geoScope locationName="Greece" geoTaxonomyURI="http://www.iso.org"
     locationCode="GR" />
  <businessEntity providerName="MyBank" providerURL="http://www.mybank.gr"
     providerAddress="Stadiou 2" providerPhoneNumber="+302107867890" />
</service>
```

Fig. 2. An example PS-WSDL advertisement

```
<USQL version="1.0" xmlns="urn:sodium:USQL">
  <USQLRequest>
    <Where>
      <Service serviceType="WebService">
        <Operation>
          <Inputs>
            <input>
              <semantics ontologyURI=" http://www.nbg.gr/LoansServicesSDO.owl">
                http://www.nbg.gr/LoansServicesSDO.owl#Customer_SSN
              </semantics>
            </input>
          </Inputs>
          <Outputs>
            <output>
              <semantics ontologyURI=" http://www.nbg.gr/LoansServicesSDO.owl">
                http://www.nbg.gr/LoansServicesSDO.owl#Credit_Score
              </semantics>
            </output>
          </Outputs>
        </Operation>
      </Service>
    </Where>
  </USQLRequest>
</USQL>
```

Fig. 3. An example USQL Request

USQL [15] is an XML-based language which enables the formulation of queries containing syntactic, semantic and QoS service requirements, allowing requesters to express their needs for heterogeneous services in a unified, efficient and consistent way. Besides the syntactic, semantic and QoS elements defined by the language, a set of operators is also provided; syntactic, semantic and QoS operators are applied to the values of the respective elements within the course of the matchmaking process. In Fig. 3, the requester asks for web services offering an operation which takes as input the social security number of the customer and returns as output his credit score.

The *heterogeneity* problem among the various approaches was addressed with the use of the layered architecture presented above and special software modules called

Mediators. A Mediator is responsible for transforming the syntactic, semantic and QoS information of a PS-WSDL advertisement or USQL query into appropriate Registry structures. Service discovery results, deriving from Registries, are also transformed reversely. Implementation of the transformation operations depends on syntactic and semantic conventions used by the Registry, as well as the support for QoS characteristics. Therefore, a distinct Mediator service is needed for every type of Registry participating in the PYRAMID-S framework, which may result in having more than one Mediators in each Gateway. Furthermore, in case a Registry uses its own domain ontology (RDO in PYRAMID-S terms) instead of the standard ontology for the domain (SDO in PYRAMID-S terms) the Mediator is responsible for transforming the concepts used for annotating the service advertisement/query from the one ontology to the other. The main benefits accruing from the use of Mediators are the following:

- They allow the participation in PYRAMID-S of Registries that do not conform to SDO.
- They provide support for reasoning. Reasoning support is very important for the matchmaking process since exact matches between service advertisements and service requests are unlikely. For example, a bank may publish a service as "LoanService" and a request issued by a client may refer to "MortgageLoanService". The advertisement and the request differ, but the matchmaking process should be able to recognize a match based on the fact (stated, for example, in an ontology for Banking services) that a loan may be a mortgage loan.
- They alleviate the service publishers/requesters from the burden of holding technical and protocol-specific knowledge about Registries; thus, they can focus on the formulation of service advertisements/queries expressed in PS-WSDL/USQL, in a consistent and rich way.
- They render PYRAMID-S extensible as it can accommodate various types of Registries through the accommodation of the appropriate Mediators.

3 PYRAMID-S Functionality and Design

The users of PYRAMID-S are either humans or agents (software programs). User actions are directed to any peer of the Gateway layer, which then, depending on the user action, interacts with either a Router peer or both a Router peer and one or more Registry peers and returns a reply. User actions may vary from service queries or advertisements to modifications of the Domain Classification Ontology (DCO), depending on the role of the user. PYRAMID-S distinguishes three different types of users:

- Simple users, who publish and discover web services
- Registry operators, who can add/remove Registries from PYRAMID-S and update Registry information
- Domain administrators, who can update the DCO by adding new domains

Fig. 4 depicts a UML use case diagram that shows the main functionality of PYRAMID-S and the three types of users who interact with it. Browsing and searching through the DCO and the SDOs, represented by the Ontology Navigation use case, is available to all types of users and greatly facilitates the rest of their actions as

it will be apparent later on. These actions are represented by the rest of the use cases depicted in Fig. 4 and are described in the following paragraphs.

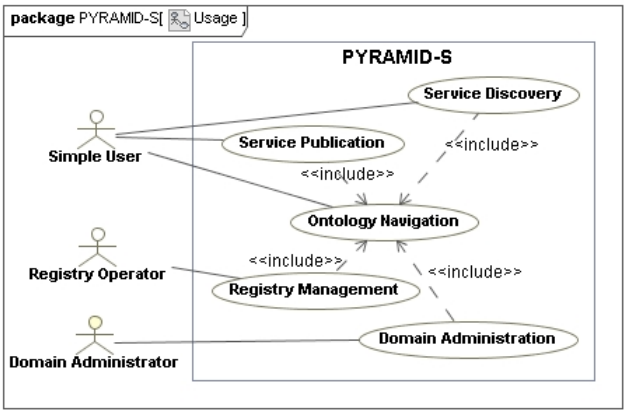

Fig. 4. Use Case diagram for PYRAMID-S

Service Publication. When a user wishes to register a service in PYRAMID-S, he or she contacts a Gateway peer and first selects the domain for publication. Afterwards, the service publisher loads the WSDL file describing the service, annotates it with the concepts of the SDO corresponding to the selected domain and then saves the resulting PS-WSDL file. Then, the publisher may designate the Registries for publication either by selecting all or several of the Registries of the previously selected domain or by selecting the Registries of a specific provider in the domain. Finally, the publisher proceeds with the publication of the PS-WSDL to the selected Registries. Then, the Gateway contacts the appropriate PYRAMID-S components in order to complete the service publication. Finally, the user is informed about the result of his/her request.
Service Discovery. When a user wishes to search for a service in PYRAMID-S, he or she contacts a Gateway peer and first selects the domain for discovery. Afterwards, the service requester specifies his/her requirements by using concepts of the SDO corresponding to the selected domain. This results in a USQL query. Then, the requester may designate the Registries for discovery. Finally, the requester proceeds with the service discovery in the selected Registries based on the USQL. Then, the Gateway contacts the appropriate PYRAMID-S components in order to complete the service discovery. The results of the Registries are then returned to the Gateway and presented to the user.
Registry Management. Registry operators may use the respective interface provided by a Gateway in order to insert/delete a Registry or update its associated properties in the PYRAMID-S system. The part of the DCO that depicts the relationships among domains is presented to the Registry operator (in a tree structure) in order to associate his/her Registry to the appropriate domain. The user input is translated into one of the following operations on the DCO:

- Insert(T_x): Based on the user input, Registry R_x is related to domain D_x and provides its services with A_x properties (T_x=<R_x, D_x, A_x>). This operation is valid

only if there is no $T_y \in DCO$: $R_y=R_x \wedge D_y=D_x$. After the completion of the operation, DCO is DCO+$\{T_x\}$.

- Delete(T_x): Registry R_x is no longer related to domain D_x. This operation is valid only if $T_x \in DCO$. After the completion of the operation DCO is DCO–$\{T_x\}$.
- Update(T_x,A_x'): The properties of Registry R_x for domain D_x are updated to A_x'. This operation is valid only if $T_x \in DCO$. After the completion of the operation DCO is DCO–$\{T_x\}$+$\{(R_x,D_x, A_x')\}$.

Domain Administration. Domain administrators may update the DCO with the addition of new domains. Domain renaming or deletion in the DCO are not allowed, as they would introduce inconsistency regarding registered Registries and services.

The aforementioned functionality is offered through a service-oriented design and implementation meaning that the functionality is provided through the definition of a number of web services. This decision entails several benefits, such as separation of concerns, ability to substitute existing web services with new ones providing improved functionality and reusability of the functionality provided by the PYRAMID-S web services from within other applications that need to perform service publication and/or discovery.

4 Conclusions

The research presented contributes to the field of web service publication and discovery. Specifically, this research focused on the confrontation of the following problems of current approaches: (a) low precision and recall in service discovery, (b) limited scalability to large number of services, service publishers and service requesters and (c) heterogeneity among the approaches. The proposed framework, which is called PYRAMID-S, addresses the above problems by categorizing Registries to domains and by functioning as a meta-Registry that controls and supports access to the Registries.

The measurements that were conducted with the help of the prototype that was implemented in the scope of this research revealed encouraging and positive results. Specifically, the experimental results have shown that the proposed system scales well as the number of involved Registries increases and that it manages to solve the disadvantages of the current approaches without incurring considerable overhead in service publication and discovery. Furthermore, the results have shown that the use of semantic and QoS information in PYRAMID-S provides high precision and recall by substantially improving on naïve keyword-based search.

The originality of this research in relation to current approaches consists in the confrontation of the heterogeneity among them. In this context, the results of this research bring valuable help to organizations or to federations of organizations that consist of a number of autonomous and heterogeneous registries; thus, enabling easier intra- and inter-enterprise integration.

PYRAMID-S supports the formation of registry federations as follows. It allows *data distribution* with the help of DCO. It supports *semantic heterogeneity* in the federation with the help of Mediators and the mappings from SDOs to RDOs. Furthermore, it supports *structural heterogeneity*, as it supports various types of Regis-

tries through the use of Mediators. Furthermore, it supports *design autonomy* as different Registries can have different algorithms for semantic publication and discovery. It also supports *execution autonomy* as Registries in PYRAMID-S can be accessed in a standalone manner without using any PYRAMID-S components.

References

1. Baresi, I. and Miraz, M.: A Distributed Approach for the Federation of Heterogeneous Registries. In: 4th International Conference of Service-oriented Computing (ICSOC'06), Chicago, USA, 2006. Lecture Notes in Computer Science, volume 4294, pp. 240-251
2. Bernstein, P. A. and Melnik, S.: Model Management 2.0 - Manipulating Richer Mappings. In: SIGMOD 2007, 11-14 June 2007, Beijing, China, 1-12
3. Choi, N., Song, I. and Han, H.: A survey on ontology mapping. In: SIGMOD Rec. 35, 3 (Sep. 2006), 34-41
4. Doan, A., Madhavan, J., Domingos, P. and Halevy, A.: Learning to map between ontologies on the semantic web. In: 11th International Conference on World Wide Web, Honolulu, Hawaii, USA, May 07 - 11, 2002, 662-673
5. Haller, A., Cimpian, E., Mocan, A., Oren, E. and Bussler, C.: WSMX - a semantic service-oriented architecture. In: International Conference on Web Services (ICWS 2005), Orlando, Florida (USA), July 2005, IEEE Computer Society, Washington, DC, pp. 321-328
6. ISO 3166 Country Codes, Available online at http://www.iso.org/iso/en/prods-services/iso3166ma/index.html
7. North American Industry Classification System (NAICS), Available online at http://www.census.gov/epcd/www/naics.html
8. OASIS ebXML Registry TC: ebXML Registry Information Model (RIM) v3.0, May 2005
9. OASIS ebXML Registry TC: ebXML Registry Services Specification (RS) v3.0., May 2005
10. OASIS UDDI Specification TC: Universal Description, Discovery and Integration v3.0.2 Specification, Feb. 2005
11. Pilioura, T., Tsalgatidou, A. and Kapos, G. D.: Specification of PS-WSDL". Technical Report, Available online at: http://www.di.uoa.gr/~thomi/TR/PSWSDL.pdf
12. Srinivasan, N., Paolucci, M. and Sycara, K.: An Efficient Algorithm for OWL-S based Semantic Search in UDDI. In: First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004) San Diego, CA, USA, July 2004, Springer Berlin / Heidelberg, pp. 96-110
13. Syeda-Mahmood, T., Shah, G., Akkiraju, R., Ivan, A., and Goodwin, R.: Searching service repositories by combining semantic and ontological matching. In: IEEE International Conference on Web Services (ICWS'05), Orlando, Florida, USA, July 2005, IEEE Computer Society, Washington, DC, pp. 13 – 20
14. Tian, M., Gramm, A., Ritter, H. and Schiller, J.: Efficient selection and monitoring of QoS-aware web services with the WS-QoS framework. In: IEEE/WIC/ACM International Conference on Web Intelligence, Beijing, China, September 2004, pp. 152-158
15. Tsalgatidou, A., Pantazoglou, M. and Athanasopoulos, G.: Specification of the Unified Service Query Language (USQL). Technical Report, Available online at: http://cgi.di.uoa.gr/~michaelp/TR/usql-1.0-spec.pdf
16. Tsalgatidou, A. and Pilioura, T.: An Overview of Standards and Related Technology in Web Services. In: International Journal of Distributed and Parallel Databases, Special Issue on E-Services, 12(2), Sep 2002, pp. 135-162
17. Verma, K., Sivashanmugam, K., Sheth, A., Patil, A., Oundhakar, S., and Miller, J.: METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and

Discovery of Web Services. In: Journal of Information Technology and Management, Special Issue on Universal Global Integration, Kluwer Academic Publishers, Vol. 6, No. 1, 2005, pp. 17-39
18. W3C Working Group: Web Services Description Language, Available online at: http://www.w3.org/TR/2007/WD-wsdl20-primer-20070326/