지능정보연구 제15권 제3호 2009년 9월(pp.15~35)

A Survey on Public Web Service Repositories

Yousub Hwang
Department of Business Administration,
College of Business Administration, University of Seoul
(yousub@uos.ac.kr)

Web service Technology has been developing rapidly as it provides a flexible application-to-application interaction mechanism. Several ongoing research efforts focus on various aspects of Web service technology, including the modeling, specification, discovery, composition and verification of Web services. The approaches advocated are often conflicting-based as they are differing expectations on the current status of Web services as well as differing models of their future evolution. One way of deciding the relative relevance of the various research directions is to look at their applicability to the currently available Web services. To this end, we conducted a survey on currently publicly available Web service repositories. Our aim is to get an idea of the number, complexity and composability of these Web services and see if this analysis provides useful information about the near-term fruitful research directions.

Received : July 17, 2010 Revision : July 25, 2010 Accepted : August 04, 2010 Corresponding author : Yousub Hwang

1. Introduction

With the rapid development of the internet technology, the World Wide Web is being used more and more in application to application communication beyond the current human-machine interaction. Web service Technology has received much attention in the last few years as it aims to provide flexible machine to machine interaction mechanism over the web. Web services ardized protocols are viewed as the potential fundamental infrastructure for the future web oriented distributed computation. The academic and industry research efforts have proposed many standards to formalize many aspects of Web service technology, including communication, invocation, monitoring, discovery and composition of Web services (Srivastrava and Koehler, 2003; Benatallah et al., 2003; Thakkar et al., 2004).

There are currently many directions in the research stream of Web service technology. The directions pursued are often conflicting-based as they are on differing expectations on the current status of Web services as well as differing models of their future evolution. Some implicitly assume that primarily applications of Web services are likely to be on the public web, while others assume that most applications of Web services are likely to be in intra-corporate scenarios. The assumptions do affect the research directions pursued. For example, those considering public Web services assume that it is infeasible to expect machine-interpretable service descriptions on the public web. They thus tend to focus on discovery and composition given just syntactic descriptions (Srivastrava and Koehler, 2003; Benatallah et al., 2003; Thakkar et al., 2004). In contrast, those looking at intra-corporate Web services expect complex, access-restricted but well-annotated services, and focus on such as automated or semi-automated composition, verification and monitoring of services (Srivastrava and Koehler, 2003; Thakkar et al., 2004).

One way of deciding the relative relevance of these research directions is thus to investigate to what extent the current ground reality of the Web services conforms to the assumptions made. To this end, we decided to conduct a survey of the existing public Web service repositories. Our research objective is to get an idea of the number, complexity, and composability of publically available Web services. The main contribution of this paper is to describe the results of our survey and discuss its implications.

We will start by providing a brief description of the current research directions in Web services. We will then describe the methodology we have used to conduct a survey on the public Web service repositories. We first describe the details of how we collect Web services from a large number of repositories, removed duplicated and validated the services. We discuss the implications and lessons of statistical analysis for the research in Web service repositories. Finally, we conclude the paper in Section 5.

2. Overview of Current Research Directions in Web Services

Web services are software components distributed on the Internet. They are accessible through the standard web communication protocols such as HTTP. The invocation of Web service is done by platformindependent and language neutral message exchange between the client and server, which makes Web services differ from other distributed computation models such as RPC (Remote Procedure Call) and makes Web services a more flexible infrastructure to build web oriented and inter-enterprise applications. This loosely coupled environment, together

with the possible service registration facilities, increases the potential of dynamic combination of existing services together.

Many standards have emerged recentlyto facilitate Web services at the level of communication (i.e., Simple Object Access Protocol ; SOAP), description (Web service Definition Language ; WSDL, Ontology Web Language for Services; OWL-S), composition (Business Process Execution Language for Web service; BPEL4WS) and discovery (Universal Description Discovery, and Integration ; UDDI).

- SOAP defines the Extensible Markup Language (XML) serialization for typed data and provides a XML serialization for typed data and provides a XML envelop for messages exchange between client and server. This is the lowest level of service invocation specification.
- WSDL is a grammar that describes Web serviceas communication endpoints capable of message exchanging. The interfaces of the operations and invocation ground information are specified in the WSDL files of services as parts of the service profiles to be published in the service registry.
- BPEL4WS is an XML based work flow definitionlanguage which describes how individual services are connected to join a business process. BPEL4WS provides rich controlstructures to combine primitive activities, such as invocation of an individual service, into complicated business logic.
- OWL-S provides Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. OWL-S market of Web services will facilitate the automation of Web service tasks including automated Web service discovery, execution, interoperation, composition and monitoring.
- UDDI provides a standard way for publishing and discovering a standard way for publishing and discovering information about Web service. A UDDI registry provides the facilities for the service providers to advertise their services in some standard industry taxonomy and also for the user to query the desired service profile.

One of the most popular problems in Web service technology addressed by both industry and academia are service discovery. At an abstract level these efforts could be classified into three main trends : one is the denotational semantic method, which describes functionality of

Web service in terms of input/output parameters and pre-/post condition for execution. Another promising method for facilitating Web service discovery is the information retrieval method that treats WSDL documents as documents containing information in the same way as books or hypertext documents. A third approach is the descriptive method that organizes Web services into predefined Web service taxonomies (e.g., subject categories or pre-enumerated keywords). The keywords that describe the Web service are organized by means of facets (possibly orthogonal), thus defining a multi-dimensional search space where each facet corresponds to a dimension.

2.1 Denotational Semantic Method

The denotational semantic method considers a service to be a function that requires inputs and generated outputs. Each service can be described by signature and specification pair; a signature represents the structureof a component's input/output parameters, whereas a specification describes a component's dynamic behavior, such as a pre-/post condition. Purtilo and Altee propose a signature matching method for the interface adaption of software components by specifying their model's parameters (Purtilo and Altee, 1991). Zaremski and Wing formally define a set of concepts related to signature-and specification matching for retrieving software components from software library (Zaremski and Wing, 1995; Zaremski and Wing 1997).

The limitation of UDDI is its lack of an explicit representation of the capability of Web services. The result is that UDDI supports the location of essential information about a particular Web service, once the Web service is known to exists, butit is impossible to locate a Web service based on what it does. Thus, there is a great need for standardized vocabulary to represent service specifications. The W3C Web Ontology Working Group defined the OWL-s, which is aimed at being a standardized broadly accepted ontology language to represent service specification. To leverage on OWL-S, Paolucci et al., propose a translation function from UDDI registry to OWL-S profiles (Polucci et al., 2002A). They adopt OWL-S as the service description language, and then discuss a matching algorithm between advertisements and requests described in OWL-S that recognizes various degrees of matching (e.g., exact, plug-in, and subsume) (Polucci et al., 2002B).

Gonzalez-Castillo et al., focus on defining the matching process based on the Description Logics (DL) subsumption relationship between the advertisement and the request (Gonzalez-Castillo et al., 2001). The authors assume an extensive representation of Web services to specify the type of service. Similar to the OWL-S Matchmaker, they define a number of degrees of

matching (e.g., exact, sub-concepts of, and subsume).

Li and Horrocks assume an intensive representation of Web services capabilities that is equivalent to the OWL-S service profile (Li and Horrocks, 2004). A matching process utilizes a modified version of the OWL-S profile to facilitate the subsumption process, and it assumes multiple degrees of matching. The only difference is the use of intersection as an additional degree of match.

Gao et al., propose a new lightweight capability description language (SCDL) to describe, advertise, request, and match Web service capabilities precisely (Gao et al., 2002). A Web service is defined by the following: name, ontological description, type, input/output parameters, and pre- and post-conditions. SCDL defines the four types of atomic Web service capability matches as follows: exact match, plug-in match, relaxed match, and not relevant. The pre- and post-conditions need some elaboration.

Benatallah et al., developed an interesting matching algorithm that utilizes both signature and specification matching (Benatallah et al., 2005; Benatallah et al., 2003). Their hybrid algorithm has a distinct feature in that it tries to retrieve a service or a collection of services that provides as many of the outputs of service request as possible and requires few inputs, which are not provided in the service request. However, while the extension with specification matching allows for a comparison between various behavioral descriptions provided by service providers accurately reflect the components' capabilities. Consequently, the search results may contain too many semantically irrelevant Web services.

Gannod and Bhatia propose a toolset that exploits a signature matching method as ameans for facilitating Web service discovery (Gannod and Bhatia, 2004). In their toolset, a service consumer generates a signature for a Web service query request by specifying a structure of input/output parameters. The toolset then compares the signature of the request with the signature of the services published in a repository. Since signature matching is based solely on structures (i.e., data type and number of input/output parameters), it is less helpful when searching for Web services on the basis of what they do (i.e., the capability of the Web service).

Cardoso and Sheth propose a novel Web service discovery method where WSDL documents are annotated according to shared process ontologies (Cardoso and Sheth, 2003). The annotation-based approach minimizes the expensive migration process from WSDL to OWL-S and allows for semantics-based Web service discovery. The authors introduce a similarity function to identify similar entity classes by using a matching process over synonym sets, semantic neighborhoods, and distinguishing features that

are classified according to parts, functions, and attributes. However, the shared process ontologies can be used effectively only in limited domains and they cannot scale up to the whole Web.

2.2 Information Retrieval Method

Traditional information retrieval methods represent each document (and user query), written in a natural language as a set of keywords called “index terms” and use the index terms to compute the degree of similarity between a document and a user query. Hence, the most important of the tools for information retrieval is the index-a collection of terms with pointers to places where information about documents can be found (Dong et al., 2004A). In an effort to increase the precision of service discovery without involving an additional level of semantic markup, several approaches based on machine-learning techniques are proposed (e.g.Sabou and Pan, 2007; Dong et al., 2004B; Heß et al., 2004; Kokash et al., 2006). All of them report enhancements in precision of automated service matchmaking.

Wang and Stroulia propose a Web service discovery method that combines information retrieval techniques with a WSDL structure matching algorithm (Wang and Stroulia, 2003A; Stroulia and Wang, 2005). To measuresimilarities between Web services, the WordNet lexicon was employed. According to the experimental results, the methods are neitherprecise nor robust. The main drawback, in our opinion, is that the methods use poor, unnormalized heuristics (e.g. matching scores of 5 or 10) in assigning weights for term similarity. The actual contents of WSDL files tend to be highly varied given the use of synonyms, hyponyms, and different naming rules. They might even not be composed of proper English words (i.e. abbreviations). Therefore, applying lexical references, such as WordNet, is not feasible. Furthermore, WordNet tends to generate an excessive number of synonyms, and thus, there were many false correlations that might affect the relatively low precision rate. The authors ignore the standard stemming process, which improves recall by reducing all forms of a term to single stemmed form. This may explain why the authors obtained relatively low recall rates in their experiments.

In (Wu and Wu, 2005), Web service similarity is defined as combining a WordNet-based lexical similarity and structural similarity. The authors aim at grounding theservice matchmaking process on a lightweight semantic comparison of signature specifications ina WSDL file. The authors add Quality of Service (QoS) to the properties of the Web service conceptual model. To measure QoS, the authors introduce four concrete parameters, such as, time to process, time to delay, time to repair, and time to failure. In order to measure service similarities between

Web services, the authors apply WordNet-based lexical reference. Thus, the authors'approach has the same flaw as that of Wang and Stroulia.

The vector space model (VSM), recognized as one of the most popular information retrieval techniques, was proposed by (Salton et al., 2005). In the VSM of information retrieval, queries and documents are represented as vectors in a high dimensionalspace where each dimension represents a word. Platzer and Dustdar implemented a VSM-based search engine for Web services (Platzer and Dustdar, 2005). The authors extracted keywords from a proportion of WSDL files, such as Endpoint URL, Message, and textual descriptions. The search engine has the capability of handling natural language based keyword and can return a list of search results with similarity rating scores. The authors do not clearly address why other contents in WSDL files (e.g. operation names and parameter information) are not considered as input features. According to other research studies (Dong et al., 2004A; Kokash et al., 2006; Stroulia and Wang, 2005), both the operation and types elements in WSDL files are important sources for inferring the capabilities of Web services. Therefore, the authors may need to expand their keyword extraction policy in WSDL files.

Fan and Kambhampati provide a survey of publicly availableWeb services in real-world public service repositories by utilizing cluster analysis. In (Fan and Kambhampati, 2005), the hierarchical clustering method was used for cluster analysis. The authors compute the similarities between services by applying VSMbased term frequency analysis based on the textual description and documentation fields of the WSDL files. After Fan and Kambhampati gathered WSDL filesfrom the public Web service registries, they removed the duplicates by using a combination of service name and provider name as the key and checking the duplicates based on the keys.

2.3 Descriptive Method

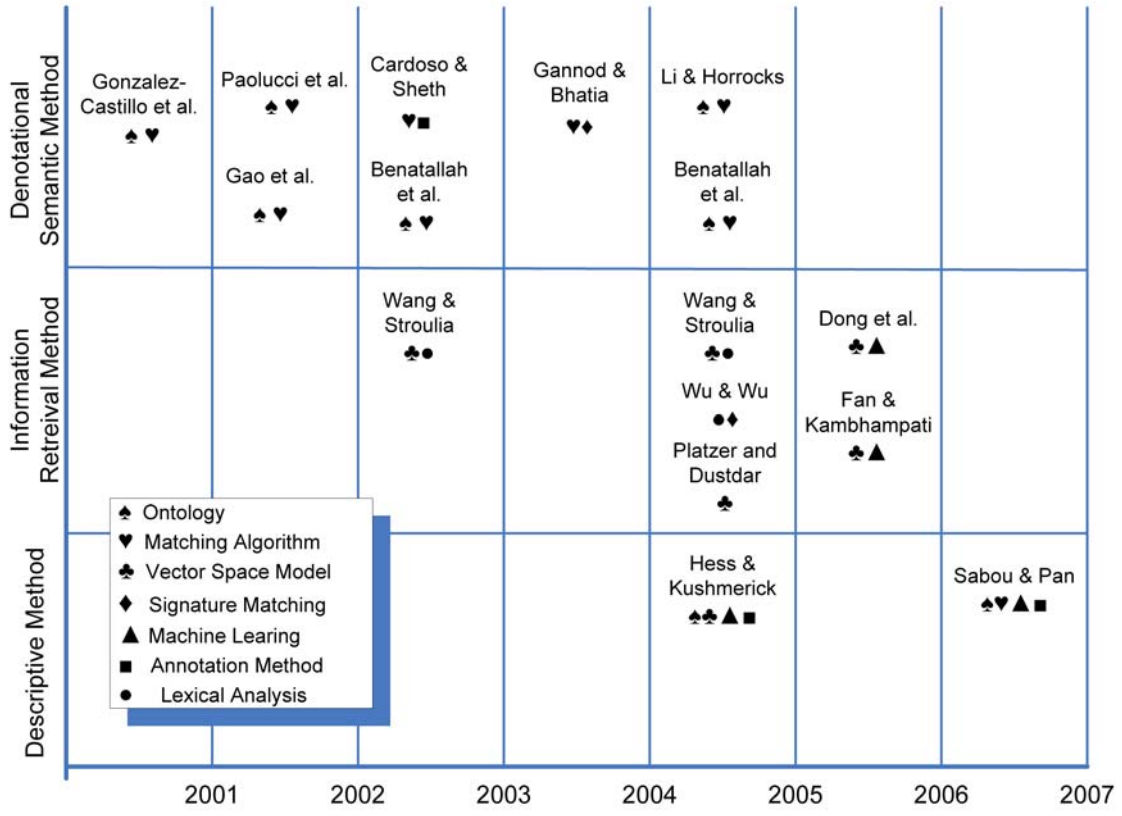
Although reported to be very effective in retrieving software components for reuse, the descriptive methods are labor-intensive for constructing predefined taxonomies (Hendler et al., 1992). To address such an issue, Heß et al., apply machine-learning techniques to generate predefined taxonomies for Web services (Heß et al., 2004). The authors aim at providing a semi-automated approach that makes use of the supervised classification and the hierarchical clustering method to suggest OWL-S based Web service taxonomies. Our work differs in that we are performing an unsupervised artificial neural network-based service similarity assessment, rather than a supervised classification. The authors assume three levels of OWL-S based taxonomies : a general taxonomy representing the nature of the services, a domain taxonomy repre

senting a collection of functionalities, and a data-type taxonomy representing a collection of semantic data categories. Because they rely on the domain-specific ontologies for Web service taxonomies, the authors'method has the same flaw as the denotational semantic methods do: the cost of managing multiple domain-dependent ontologies is too high.

Sabou and Pan propose an ontology learning-based approach for (semi-) automatically concept identification, which will extend the predefined Web service taxonomy, thus ensuring that the predefined Web service taxonomy reflects the contents of the underlying Web services (Sabou and Pan, 2007). The newly extracted concepts can be utilized as a set of pre-enumerated keywords for service discovery. The authors observed that most of the noun phrases in the textual descriptions denoted the parameters of the service while the verbs indicated the functionality of the service. The ultimate objective of Sabou and Pan's research is generating domain-specific ontologies in a bottom-up fashion (learned from available sources) rather than being imposed in a top-down fashion (requiring costly manual generation of ontologies).

In this section, we examined approaches based on descriptive methods, whose primary method was the semi-automatically generating predefined taxonomies for Web services by utilizing machine-learning techniques. One drawback of the existing descriptive method-based Web service discovery research is relying on OWL-S based domain-specific ontologies, which are not widely adopted and require expensive migration processes for existing WSDL files. Furthermore, the proliferation of domain-specific ontologies may cause an ontologies management problem, which is not supported in the existing OWL standard. Our approach differs in that we are generating domain-independent Web service taxonomies by learning from available sources, rather than domain-specific Web service taxonomies, such as OWL-S based Web service taxonomies. We have presented a summary of the literature on service discoveries classified according to three major methods focused on facilitating service discovery. The service discovery proposals we reviewed are plotted on a timeline (see <Figure 1>).

The underlying assumption of these approaches is that there are well-defined domain ontologies and the services are marked up properly with those ontologies. The process of reasoning with the ontologies would then help locate the services with desired functionalities and properties. The main question here is how feasible is it to expect semantically marked up services.



<Figure 1> Summary of Web service Discovery Research

An important way to decide which of the approaches is more relevant will be to have an idea on what type of application will Web services support in the near future. There are two diverging views here-some argue that Web services will find more use in intra-corporate scenarios. In such cases, it is likely that services will be annotated by the providers using a consistent ontology. Service discovery is likely to be less of a challenge and supporting non-trivial service composition is feasible.

Others see the main role for Web services to be on the public web-with multiple services being available to lay users. In this case, the dream of consistent semantic annotation seems less feasible. We are likely find mostly free text annotations of services, making automated service discovery, and the attendant extraction of semantic from syntactic descriptions, a pressing problem.

While we cannot gauge the use of Web services in intra-corporate scenarios, it is possi-

ble to conduct a survey on public Web service repositories. This is what we do here-in the hope of that it will shed light on what models of evolution of Web services are closer to current reality.

3. Survey on Public Web Service Repositories

By conducting a survey of the public Web services we wanted to see (1) how many public Web services were there, (2) how complex they were and (3) how meaningfully they were documented.

At first glance, getting a snapshot of what services are actually available on the public web would seem easy because we have the UDDI repositories promoted by many leading industry organizations. But the truth is, the current UDDI is still evolving and not very mature. There is no mechanism of verification or business model that could enforce the service providers to only register services that are well implemented and ready to be understood and integrated to user applications. In fact, current UDDI repositories such as *uddi.ibm.com* allow anybody to register almost anything as a Web service entry, and when we looked into the registries, a very large portion of those registered services were either “hello-world”-style simple testing or experimental services or not actual services at all.

Many of the repository entries do not even have a valid WSDL file URL, let alone the actual end point of the services. So obviously the UDDI registries are not good start for us to have a good picture of what services are available online. There are some other major online Web service repositories though, which do not necessarily conform to UDDI standards and do not yet have very large number o f registered entries, but these registries have much higher percentage of services registered that are actually available. We took a comprehensive study on the web and found several largest and most representative Web service registries, or directories. The union of the registered services on these repositories seems to cover a large portion of all the ones available online and represents their properties and features to a reasonable degree. So we took these repositories as the source of the collection of the real Web services.

To find out what services are there, we first crawled these service repositories, and then processed the data collected to remove the invalid entry and duplicates. Thenwe performed a text description and documentation based statistical analysis on the collected services.

3.1 Spidering the Public Web Service Repositories

To collect information about the current available Web services, we wrote several spider

programs to fetch the registered information of the Web services. The public Web service repositories we crawled are :

- www.xmethod.com
- www.biningpoint.com
- www.webservicex.com
- www.remotemethod.com
- www.webservicelist.com

These repositories usually have the query facilities to do the keyword lookup or category browsing on the registered information. The services registered usually have the information about the names, providers, textual descriptions and the URL of the WSDL files. We collected all this information and inaddition followed the URL and fetched the WSDL files into our relational database repository. Sometimes the URLs do not point to the WSDL files but rather to the introductory html pages of the services, in such cases we followed this kind of link and tried to find the WSDL file URL in the pointed page too. To filter the invalid registry information which is very common in all the repositories, we discarded the collected service entries which do not have a valid URL to their WSDL file or to a page that contains a URL of a WSDL file. Here we only look a the string representation of the URL to decide if it points to a WSDL file and later we further validate the collected WSDL files by utilizing WSDL parser.

3.2 Removing Invalid Entries and Duplicates

Some of the registered services might not have a valid WSDL file entry, or the WSDL file is not a well-formed xml document, or the WSDL file does not conform to the WSDL standards. We considered such entries as invalid ones. There are also a lot of duplicates among the collected service entries.

To remove the invalid entries we parsed every fetched WSDL file first to see if it is a valid XML document and eliminate the invalid ones from the relational database repository. Then for the rest, we performed a simple check of their conformance to the WSDL standard by checking the existence of several necessary WSDL tags inside the file and eliminated the invalid ones.

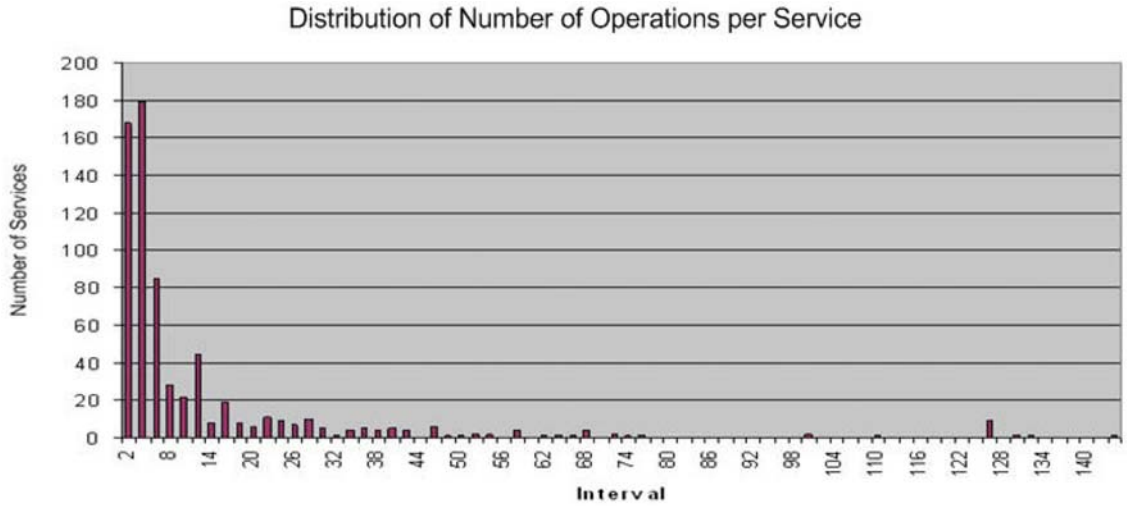
The current version of the Web services Description Language for Java Toolkit (WSDL4J) does not support the query <wsdl : types> element, which contain detailed input/output parameter information, such as parameter names, data type, minimum/maximum occurrence, enumerated data values, and so on. In order to obtain all detailed contents from the WSDL files, we implemented the WSDL parser by extending a generic XML parser. Our WSDL parser has four functionalities: retrieving the contents of

WSDL specification elements, extracting the terms from the concatenated words, preprocessing the extracted terms for further analysis, and storing the collected information into our relational database repository. To remove the duplicates, we used the combination of service description information including service name, provider's name, operation name, parameter names as the key and checked the duplicates based on the keys.

This step removed all the invalid entries and duplicates. We obtained 674 valid entries out of the total 1789 entries in the collection. Next, we performed complexity analysis of the collected Web service entries.

3.3 Complexity of the Web services

One way of measuring the complexity of the publically available Web services is to see how many individual operations are involved in the individual services. We collected the information of the number of operations in each service as the measure of the complexity of the services. <Figure 2> shows the distribution of this measure on the whole collection (of the 674 Web services). More than 66% of the WSDL files have less than 5 operations and more than 31%of them have only one operation. Moreover when we looked into the WSDL files of the services with multiple operations, more often than not the operations do not have interactions among them. Very few services have more complicated interoperation semantics (which is not explicitly defined in the WSDL file). We also tried to find some interesting composition of the services by manually checking the compatibility of the operations among these services, but it

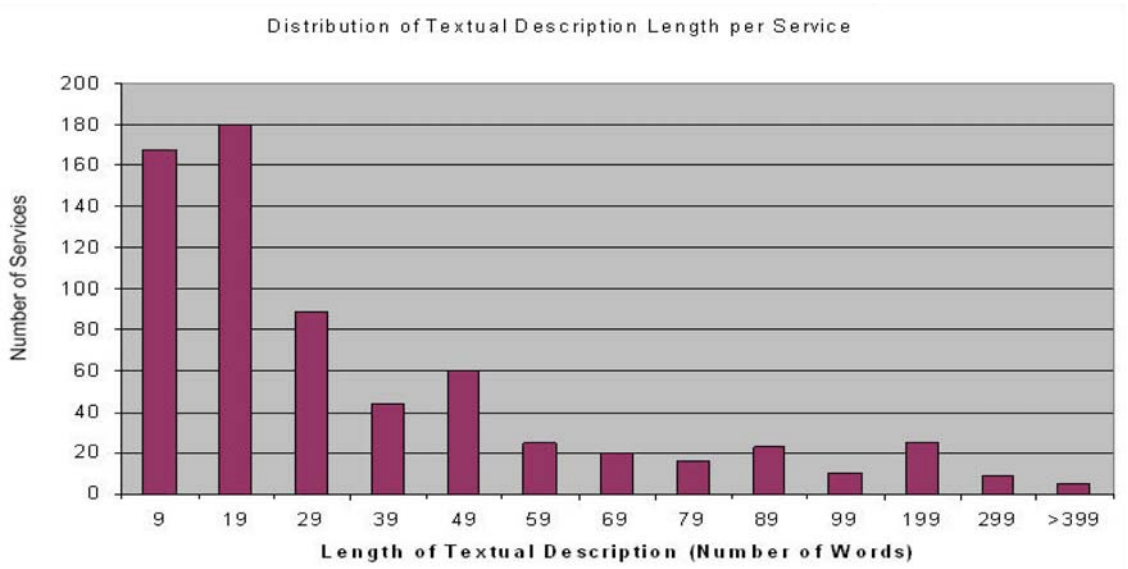


<Figure 2> Distribution of Number of Operations per Web service

turned out that no composition with more than 3 operations could be found in this collection. It seems that at least at the current stage we do not have large numbers of public Web services which are both very complicated and have potential to be composed with other services. The motivation to research of the composition of "complicated" Web service must come from intra-corporate scenarios.

3.4 Semantics of the Web services

From another point of view, given the current available Web services, if an application developer simply wants to use a service in his application, are those services ready to be used? For a developer to integrate a service into his application a key problem is to understand both semantically and syntactically about the services and the operations they support. The only way for the developers to get the semantics of the services is to read and interpret the textual description and the documentation of the services. The amount and accuracy of these textural resources directly determine if the semantics could be interpreted correctly. As stated above, these types of information are used in our data analysis and we noticed that sometimes these textual resources are not enough to semantically represent capability of Web services. One may argue that the current WSDL standards are not machine oriented and the WSDL files are supposed to be consumed by human being. However, it is questionable as to whether the service providers are seriously using the WSDL files as the way to convey the correct interpretation to the developers who will use them. To settle this, we per-



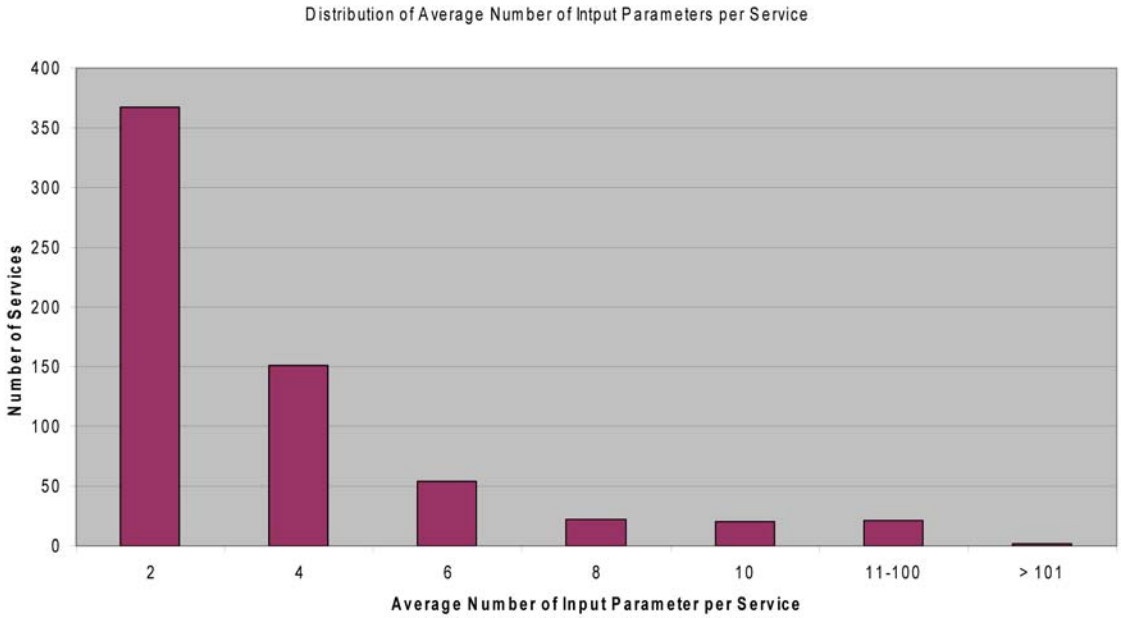
<Figure 3> Distribution of Textual Description Length per Web service

formed a statistical analysis on the available Web service registration information.

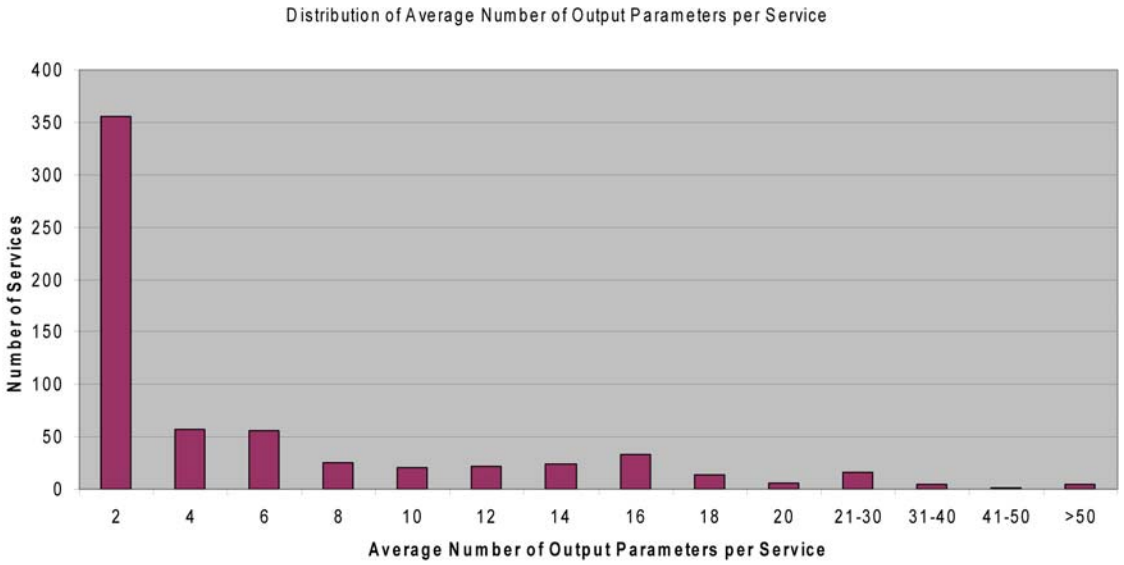
We first collected the information of the lengths of the textual description of the Web services (including the registration information and the “documentation” field of the Web service in their WSDL files) as the measure of amount of information conveyed in the service profiles. <Figure 3> illustrates the distribution of the lengths (in terms of number of words) in the collection of the 674 Web services. Most WSDL files (more than 80%) have textual description of less than 50 words, and more than 64% of them have textual descriptions with less than 20 words. Consequently, it is questionable as to whether the semantics of Web services can be described adequately with less than 20 words.

We extracted detailed parameter information (including parameter names, enumerated data values, etc.) from the collected WSDL files. We collected 113,444 parameters from 638 of the WSDL files in the collection. We failed to extracted information from the remaining 36 WSDL file because they refer to external XML table namespaces, which are not accessible through the Internet. The collected parameters consist of 36,157 input parameters and 77,287 output parameters.

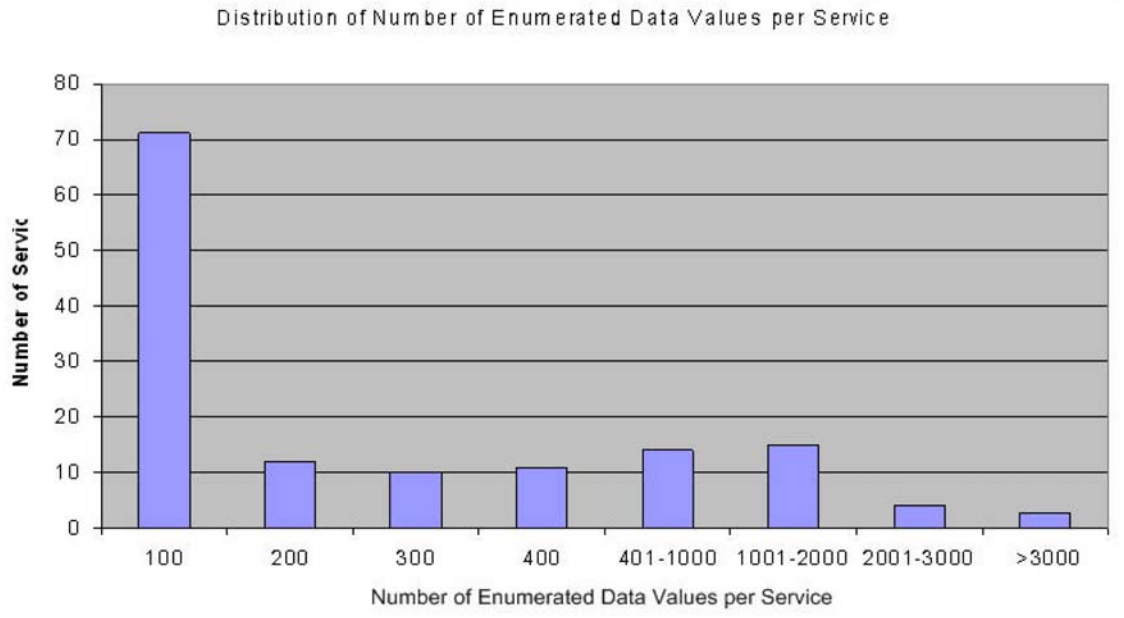
<Figure 4> depicts the distribution of average number of input parameters per service in the 638 WSDL files. As we can see, most of services (more than 81%) have less than 5 input parameters per Web service.



<Figure 4> Distribution of Average Number of Input Parameters per Web service



<Figure 5> Distribution of Average Number of Output Parameter per Web service



<Figure 6> Distribution of Number of Enumerated Data Values per Web service

<Figure 5> shows the distribution of average number of output parameter per service in the collection. Most of services (more than 80%) have less than 11 output parameters per Web

service, and more than 69% of them have less than 3 output parameters per service.

We also extracted 52,517 enumerated data values for input parameters from 137 of the WSDL files in the collection. Some of the extracted enumerated data values for a given input parameter are useful for inferring domain of activity (i.e., form the enumerated data values like 'Fahrenheit', 'Celsius', 'Kilogram', or 'SpanishTOEnglish') or geographic context (i.e., from the enumerated data values like 'Arizona', 'Florida', 'Virginia', 'Utah', or 'France') of a given Web service. As you can see in <Figure 6>, most of the Web services that contain enumerated data values for input parameters (more than 75%) have enumerated data values for input parameters less than 400 words and more than 68% of them have enumerated data values for input parameter less than 100 words.

4. Implications and Lessons Learned

From the statistical analysis above we can look again at the current directions of research in Web service from their relevance to the current Web services available in the public web. A general caveat is in order before we proceed to enumerate the lessons-it is entirely possible that we are in the stone-ages as far as publicly available Web services are concerned; and that the complexity of publicly available services will improve significantly in the near future as the infrastructure standards took root. Nevertheless, we believe that it is worthwhile to evaluate the potential fruitfulness of the current research directions in Web services from the stand point of the current survey on the public Web services.

The preponderance of data source-oriented Web services explains some extent an apparent paradox in the approaches to service composition that have been advocated. Specifically, although in theory service composition is expected to involve complex plan synthesis (Thakkar et al., 2003; Srivastava and Koehler, 2003), several projects use composition techniques that are indistinguishable from query plan formation in data integration scenarios (Thakkar et al., 2003; Ponnekanti and Fox, 2002; Kambhampati and Knoblock, 2003).

A lot of research efforts on Web services have concentrated on the service discovery/retrieval issue. The discovery issue is most critical for the publicly available sources. One interesting observation is that, if the text description such as WSDL and UDDI entry is the only source to describe the Web services, the simple information retrieval techniques perform well, as long as such descriptions are reasonably long. If that is the case, the problem of "discovery" itself is not likely to be a challenging one because the general discovery does not seems to be able to achieve more than whatthe current commercial search engines already do. Nevertheless the

performance of service discovery depends on not only the techniques to “discover” but also the quality of the registration information of the registered Web services themselves, which currently are not guaranteed without a proper business model to enforce and verify the service publishing activities. While an argument can be made that retrieval will be more challenging as Web services evolve and become more involved, it is also possible that the same evolution will advance the Web service repository such that there will be more structured entries on registries making retrieval easier.

We also found that there are very few ways of composing services available online, mainly because of the lack of services and the correlations among them. Most of the current available services can be viewed as data sources with interfaces clearly defined with WSDL. Data sources with proper defined XML interfaces are easier to be integrated compared to current web database integration scenario because the integrators no long need to screen-scrape the html pages to isolate the real data from the fancy representation. But when it comes to the problem of composition, it does not seem very different from the current data integration problem. Of course we have to admit that in intra-corporate scenarios there may be other types of “complex” Web services with data updates, complicated interactions and other run time semantics involved, the composition as well as the verification and monitoring, of such services would be a challenging problem for public Web services.

5. Conclusion

Web services are becoming more and more popular in both the industry and academic research. The relevant problems include the modeling, communication, composition, discovery, verification and monitoring of Web services. Prior to the research on these problems we have to know what kind of services actually exist and on the other hand from the academic research point of view we have to figure out what is the shortcoming and defects of the current Web service model and what problem should be handled as the future direction.

In this article, we presented a survey on the Web services currently available in the public web and discussed the relevance of various research issues of Web service technology based on the data and statistics collected. We found that there is a big gap between the frontier research activities and the reality of the Web services. We also argued that the problem of discovery is not a very feasible one given the syntactic specification and textual description of services as the basis. In addition, we found that most current services can be viewed as data sources using WSDL to describe the interfaces and the

composition of such services may well be challenging problem, but the motivating scenarios are not likely to come from current public Web services. We also found that the current WSDL standards are used more often for the documentation purpose rather than clearly defining the syntax and semantics of the Web services which is inadequate to be easily used by the application developers and the research on automated or semi-automated annotation of services would be a challenging issue.

In closing we would like to reiterate that all our observations and conclusions are based on the Web services publicly available on the web. It would be interesting to do a similar study on the current status of the intra-corporate Web services. Intra-enterprise Web services and well controlled collaborative inter-corporation Web services could have characteristics that are significantly different from those of the public ones covered in our survey.

References

Benatallah, B., M. Hacid, A. Leger, C. Rey, and F. Toumani, “On automating Web services discovery”, *Journal on Very Large Data Bases*, Vol.14, No.1(2005), 84～96.

Benatallah, B., M. Hacid, and C. Rey, “Semantic reasoning for Web services discovery”, in Proceedings of the Workshop on E-Service and the Semantic Web, (2003).

Cardoso, J. and A. Sheth. “Semantic E-Workflow Composition”, *Journal of Intelligent Information Systems*, Vol.21, No.3(2003), 191～225.

Dong, X., J. Madhava, and A. Halevy, “Similarity search for Web services”, In Proceedings of the 30th VLDB Conference, (2004A), 373～383.

Dong, X., J. Madhava, and A. Halevy, “Mining Structures for Semantics”, ACM SIGKDD Explorations Newsletter, Vol.6, No.2(2004B), 53～60.

Fan, J. and S. Kambhampati, “A Snapshot of Public Web services”, SIGMOD Record, Vol.34, No.1(2005), 24～32.

Gannod, G. C. and S. Bhatia, “Facilitating Automated Search for Web services”, *In Proceedings of the IEEE International Conference on Web services*, San Diego, California, USA, July, (2004).

Gao, X., J. Yang, and M. P. Papazoglou, “The Capability Matching of Web services”, In Proceedings of IEEE Fourth International Symposium on Multimedia Software Engineering (MSE‘02), Newport Beach, CA, USA, Dec, Vol.11, No.13(2002), 56～63.

Gonzalez-Castillo, J., D. Trastour, and C. Bartolini, “Description Logics for Matchmaking of Services”, Technical Report HPL-2001-265, HP Labs, (2001).

Hendler, J., R. P. Diaz and C. Braun, “Computing Similarity in a Reuse Library Systems : An AI-based approach”, ACM Transactions on software Engineering and Methodology, Vol.1, No.3(1992), 205～228.

Heß, A., E. Jonston, and N. Kusherick, “Semi-atuomatically Annotating Semantic Web services”, in Proceedings of Semantic Web Conference, (2004).

Kokash, N., W. Heuvel, and V. D’Andrea, “Lever-

aging the Web service Discovery with Customizable Hybrid Matching", In Technical Report DIT-06-042, University of Trento, (2006).

Kambhampati, S. and C. Knoblock, "Guest Editors' Introduction : Information Integration on the Web", IEEE Intelligent Systems, Vol.18, No.5(2003), 14~15.

Li, L. and I. Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology", *International Journal of Electronic Commerce*, Vol.8, No.4(2004), 331~339.

Paolucci, M., T. Kawamura, T. Payne, and K. Sycara, "Semantic Matching of Web services Capabilities", *In Proceedings of the First International Semantic Web Conference*, 2002A.

Paolucci, M., T. Kawamura, T. Payne, and K. Sycara, "Importing the Semantic Web in UDDI", *In Proceedings of the First International Semantic Web Conference*, (2002B).

Platzer, C. and S. Dustdar, "A Vector Space Search Engine for Web services", *in Proceedings of the Third European Conference on Web services*, (2005).

Ponnekanti, S. R. and A. Fox, "SWORD: A Developer Toolkit for Web Service Composition", *In Proceedings of the 11th International World Wide Web Conference*, Honolulu, HI, USA, (2002).

Purtilo, J. M. and J. M. Atlee, "Module reuse by interface adaptation", *Software Practice and Experience*, Vol.21, No.6(1991), 539～556.

Sabou, M. and J. Pan, "Towards Improving Web service Repositories through Semantic Web Techniques", *Web Semantics: Science, Services and Agents on World Wide Web*, Vol. 5, No.2(2007), 142～152.

Salton, G., A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing", *Communications of the ACM*, Vol.18, No.11 (1975), 613～620.

Srivastava, B., and Koehler, J., "Web service composition－current solutions and open problems", *In Proceedings of ICAPS'03 Workshop on Planning for Web Services*, Trento, Italy, June(2003).

Thakkar, S., C. Knoblock, and J. L. Ambite, "A View Integration Approach to Dynamic Composition of Web Services", In Proceedings of ICAPS'03 Workshop on Planning for Web Services, Trento, Italy, June(2003).

Wang, Y. and E. Stroulia, "Flexible Interface Matching for Web-Service Discovery", In Proceedings of the Fourth International Conference on Service Oriented Computing (WISE'03), Rome, Italy : IEEE Computer Society Press, (2003).

Wu, J. and Z. Wu. "Similarity-based Web service Matching", *in Proceedings of the IEEE International Conference on Services Computing*, (2005).

Stroulia, D. and Y. Wang, "Structural Semantic Matching for Assessing Web service Similarity", *International Journal of Cooperative Information Systems*, Vol.14, No.4 (2005), 407～436.

Zaremski, A. M. and J. M. Wing, "Signature Matching : a Tool for Using Software Libraries", *ACM Transactions on Software Engineering and Methodology*, Vol.4, No.2(1995), 146～170.

Zaremski, A. M. and J. M. Wing, "Specification matching of software components", *ACM Transactions on Software Engineering and Methodology*, Vol.6, No.4(1997), 333～369.

Abstract

공공 웹서비스 저장소에 대한 연구조사

황유섭*

웹서비스 기술은 응용 어플리케이션 간의 원활한 상호작용 방법을 제공하기 위해 급격하게 발전하고 있다. 현재 웹서비스 관련 연구들은 웹서비스 모델링, 웹서비스 발견, 조합 그리고검증 등의 다양한 방향으로 진행되고 있다. 현재 웹서비스 기술적 상황에 대한 기대라는 관점과 웹서비스의 미래 확장 모델에 관하여 많은 연구들은 상충되고 있다. 다양한 연구방향의 상대적 적절성을 평가하는 한 방법은 현재 공공영역에 존재하는 웹서비스들에 대하여 각각의 연구방향들의 적용 가능성을 연구조사하는 것이다. 따라서, 이 논문에서현재 공공영역에 존재하는 웹서비스들에 대한 연구조사 실행하고 결과를 보고한다. 이논문의 목적은 공공영역에 현재 존재하고 있는 웹서비스의 개수와 복잡성, 그리고 조합성 정도를 측정하고자 한다. 또한, 가까운 미래의 유익한 웹서비스 연구방향에 관한 유용한 정보를 제공하고자 한다.

Keywords : 웹서비스, 시맨틱, 복잡성, 웹서비스 발견, 웹서비스 조합

* 서울시립대학교 경영대학 경영학부

저 자 소 개

황유섭

현재 서울시립대학교 경영대학 조교수로 재직 중이다. The University of Arizona에서 경영정보시스템을 전공하여 경영학사, 석사 그리고 경영학 박사학위를 취득하였다. 미국 NASA와 Raytheon의 Hydrology Resource Management Project에 연구원으로 참여하였으며 Photogrammetric Engineering and Remote Sensing과 ER 학회지, Information Systems Review, 지능정보연구 등에 논문을 게재하였다. 주요 관심분야는service-oriented computing, forecasting, artificial neural network의 활용 방안 연구, IT strategy 등이다.