Empirical Research on Search model of Web Service Repository

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

The World Wide Web is transitioning from being a mere collection of documents that contain useful information toward providing a collection of services that perform useful tasks. The emerging Web service technology has been envisioned as the next technological wave and is expected to play an important role in this recent transformation of the Web. By providing interoperable interface standards for application-to-application communication, Web services can be combined with component-based software development to promote application interaction and integration within and across enterprises. To make Web services for service-oriented computing operational, it is important that Web services repositories not only be well-structured but also provide efficient tools for an environment supporting reusable software components for both service providers and consumers. As the potential of Web services for service-oriented computing is becoming widely recognized, the demand for an integrated framework that facilitates service discovery and publishing is concomitantly growing.

In our research, we propose a framework that facilitates Web service discovery and publishing by combining clustering techniques and leveraging the semantics of the XML-based service specification in WSDL files. We believe that this is one of the first attempts at applying unsupervised artificial neural network-based machine-learning techniques in the Web service domain. We have developed a Web service discovery tool based on the proposed approach using an unsupervised artificial neural network and empirically evaluated the proposed approach and tool using real Web service descriptions drawn from operational Web services repositories. We believe that both service providers and consumers in a service-oriented computing environment can benefit from our Web service discovery approach.

Received : October 17, 2010 Accepted : November 04, 2010 Corresponding author : Yousub Hwang

1. Introduction

Enterprises in the public and private sectors have recently resulted in a surge in Web services for business, making software components available over the Internet. Web service is a pretty simple idea : standardize the generic functions that are widely used by many applications into reusable components (services) that are accessible over a network. Software reuse, defined as the process of using existing software artifacts rather than building them from scratch (Krueger, 1992), is an important strategy for improving software development efficiency and improving the quality of software systems (Basili et al., 2003; Chávez et al., 2000; Poulin et al., 1993). Many enterprise software vendors (i.e., Oracle and SAP) have moved into Web services for their new feature sets - and with good reason.

A prerequisite for reusing software components is the ability to find the right software component(s). However, service discovery is becoming problematic with the increasing number of Web services. The main objective of this research is to develop a more effective mechanism for Web service discovery. Our approach combines a clustering technique and leverages the contents of the XML-based service specification in Web Services Description Language (WSDL) files. We propose a software tool called **Web Serv**ice **S**elf **O**rganizing **M**ap (WebServSOM) using an artificial neural network as its core module to semi-automatically generate Web service taxonomies that reflect subject category context for Web services.

We will start by providing a brief overview of the current public web service repositories to illustrate problem aspects of existing search models for Web service discovery in Section 2. In Section 3 we describe our Web service taxonomy generation framework for facilitating Web service discovery. We will then explain about our experimental design in Section 4. In Section 5 we summarize our findings from human subject experiment with different search models in terms of objective performance measures and subjective performance measures. Finally, we conclude the paper in Section 6.

2. An Overview of Public Web service Repositories

Sabou and Pan provide a survey of the public Web services repositories in (Sabou and Pan, 2007). The authors evaluate seven public Web services repositories with two assessment criteria : the search facility and the browse facility. We update Sabou and Pan's survey by removing the currently unavailable repositories (i.e., Binding Point, NetXML, and SalCentral) and adding newly available repositories (i.e., Remote Methods and *Woogle*). We also extend the assessment criteria based on Sabou and Pan's work. In this section, we summarize six public Web services repositories. For each, we describe the facilities that they offer to retrieve the available services and point out the problematic aspects when applicable.

2.1 Assessment Criteria

Given that we discuss a number of Web service repositories in this research, it is important to define a uniform set of criteria that we use to assess and compare these repositories. We evaluate the Web services repositories with the following two perspectives : *major methods for accessing contents* and *effort of combining searching and browsing*.

Major methods for accessing contents : In a broad sense, there are several methods for accessing the contents of the existing Web service repositories. One approach is query-based search, which allows users to enter a keyword or set of keywords that, in their opinion, best characterizes Web services that are of interest (Chen et al., 1998). The Web services repository translates this request into a query and identifies the information space for appropriate matches, which are returned. Some repositories allow the user to refine a query-based search by specifying advanced search option, such as Boolean operators ("AND", "OR" and "NOT"). Query-based search can be further classified into two different levels depending on the granularity of the term unit (TU). For instance, query-based search can be performed at the level of substring or token. The latter attempts to match a search term to a whole word only (token search), while the former retrieves Web services that contains the search term as a substring of the string (substring search). Another method is taxonomy browsing. Taxonomy is a hierarchical arrangement of thematic structure inherent in a collection of documents which can improve Internet search precision (Pahlevi and Kitagawa, 2003; Muller et al., 1999). Marchionini and Shneiderman define browsing as an exploratory, information seeking strategy that depends upon serendipity for ill-defined problems and for exploring new task domains in (Marchionini, 1987; Marchionini and Shneiderman, 1988). Because browsing is frequently used when users only have general topics or they are not sure how to narrow their search from a general topic, users typically rely on pre-existing taxonomies of information organization as they explore. We identify three types of Web service taxonomies : (i) industry/product standard classification schemes, (ii) lightweight service taxonomies (where the categories used are determined by the domain of Web services or a certain function they provide), and (iii) alphabetically ordered Web service names. A third method for accessing the contents of Web service repositories is listing, which is a list of hyperlinks for Web services.

Effort of combining searching and browsing : Browsing and searching are the two main strategies for finding information online. Manber et al. argue that by combining browsing and searching users will be given a much more powerful tool to find their way (Manber et al., 1997). A Web service repository allows users to further refine a keyword search by restricting it to a given subject category or sub-division of the entire database. The combined searching strategy is more efficient than searching the entire database but, as a consequence, users are unable to identify relevant information that may exist outside of the subject category chosen (Chen et al., 1998; Manber et al., 1997). In this assessment criterion, we check whether a Web services repository allows users to conduct search tasks by combining query-based search and taxonomy browsing simultaneously.

2.2 Evaluation Summary

Based on the overview and assessment criteria presented above, we are able to summarize the situation of public Web services repositories. We encountered three simple ways of accessing the content of Web services repositories : listing, query-based search, and taxonomy browsing. As listed in <Table 1>, one repository (i.e., *Xmethods*) that we evaluated simply presented all the available services as a long list, which is clearly insufficient for Web service discovery.

A query-based search is performed on the textual sources attached to the Web services (e.g., name, textual description, or the names of the WSDL operations). We encountered two cases where keyword term matching was performed at the token level (token match), two cases where keyword term matching was performed at the substring level (substring match), and two cases where search facilities were not supported (refer to <Table 1>). Note that a substring match leads to many hits that have no content relevance for the search. We identified only one case (i.e., *Woogle*) where advanced search options (e.g., Boolean operators with multiple keyword terms) are supported. Therefore, more effective accessing mechanisms are still needed to improve the discovery of semantically relevant Web services. <Table 2> summarizes query-based searching facilities in the public Web services repositories we evaluated.

A third extensively used access method in public Web services repositories is browsing based on different Web service taxonomies. As listed in <Table 1>, we encountered three types of Web service taxonomies : alphabetically or-

<Table 1> Summary of Public Web Services Repositories

Repository	Content accessing methods	Effort to combine multiple methods
UDDI	Substring match and Industry/product standard classification schemes	None
WebServiceX	Lightweight service taxonomy	None
WebServiceList	Token search, Lightweight service taxonomy, and Alphabetically organized Web service names	None
Xmethods	Listing	None
RemoteMethods	Substring search, Lightweight service taxonomy	Simultaneously
Woogle	Token search, Lightweight service taxonomy	Simultaneously

ganized Web service names, industry/product standard classification schemes, and lightweight Web service taxonomies. One repository (i.e., *WebServiceList*) that we evaluated offers Web service browsing via alphabetically organized Web service names, which is insufficient when the service consumer does not know the exact name of the Web service for which he/she is looking.

Another type of Web service taxonomies that we identified is industry/product standard classification schemes (i.e., NAICS or UNSPSC). One of the most significant issues in industry/product standard classification schemes is that these schemes are often under-populated; several of their categories contain no or few Web services. A second major issue is that there is no guarantee that there is consensus among service providers and consumers about industry/product standard classification schemes. The third major issue is that industry/product standard classification schemes require both service providers and consumers to have prior knowledge of the service classification schemes. In particular, if service consumers are not familiar with the industry/product standard classification schemes, they usually cannot get satisfactory retrieval results (Zhuge and Liu, 2004). Finally, it is difficult for industry/product standard classification schemes to represent the underlying functionality of Web services.

The third type of Web service taxonomies that we identified is lightweight service taxonomies. Unlike the industry/product standard classification schemes, they have only a few top categories (a maximum of 17), which, in most cases, are not further specialized. Many categories are over-populated with instances, and there is a need to extend the set of categories using new terms as the underlying data set evolves. The lightweight service taxonomies are qualitatively poor. In particular, the scope of the lightweight service taxonomies is often ambiguous since their categories often correspond to different category schemes. For instance, some describe domains of activity (e.g., business and economy) while others describe types of functionality (e.g., validation). In addition, it is often unclear how the categories are created and populated with instances. Web

<Table 2> Summary of Query-based Search in Public Web Services Repositories

Repository	Query-based search	
	Level of keyword term matching	Usage of advanced search options
UDDI	Substring match	None
WebServiceX	None	None
WebServiceList	Token match	None
Xmethods	None	None
RemoteMethods	Substring match	None
Woogle	Token match	Boolean operators

services repositories seldom use Web service taxonomies due to the cost of acquiring and maintaining them. The high cost is mainly due to the size of Web services repositories. They often contain a couple hundred services. Building extensive and balanced Web service taxonomies for these services requires a considerable amount of manual effort. Therefore, there is a need for an integrated framework that (semi-)automatically generates a well-balanced Web service taxonomy reflecting both the content and the functionality types provided by the Web services. <Table 3> summarizes the different characteristics of taxonomy browsing in the public Web services repositories.

As listed in <Table 1>, we encountered only two cases (*RemoteMethods* and *Woogle*) that attempt to combine searching and browsing simultaneously to improve the precision of the search results, two cases (*UDDI* and *WebServiceList*) that support both searching and browsing separately, and two cases (*WebServiceX* and *Xmethods*) that support either a simple list of Web services or lightweight service taxonomy-based browsing only.

Consequently, the situation of public Web services repositories can be summarized as follows :

1. Simple content accessing methods are used; two repositories support token-level query-based search, one repository provides advanced search options, and two repositories allow users to combine searching and browsing simultaneously.
2. Insufficient Web service taxonomies are used for browsing Web services; large industry/product standard classification scheme-based Web service taxonomies are too large for both service providers and consumers, while lightweight Web service taxonomies are ambiguously created and populated with instances.
3. There is a lack of an integrated framework for generating a Web service taxonomy that reflects both the content and the functionality types provided by the Web services.

<Table 3> Summary of Taxonomy Browsing in Public Web Services Repositories

Repository	Taxonomy Browsing			
	Type of taxonomy	Scope of taxonomy	Size of taxonomy	Depth of taxonomy
UDDI	Standard classification schemes	Industry/product sector	16,000	5
WebServiceX	Lightweight	ambiguous	7	1
WebServiceList	Lightweight	ambiguous	17	1
Xmethods	None	None	None	None
RemoteMethods	Lightweight	ambiguous	45	2
Woogle	Lightweight	ambiguous	42	3

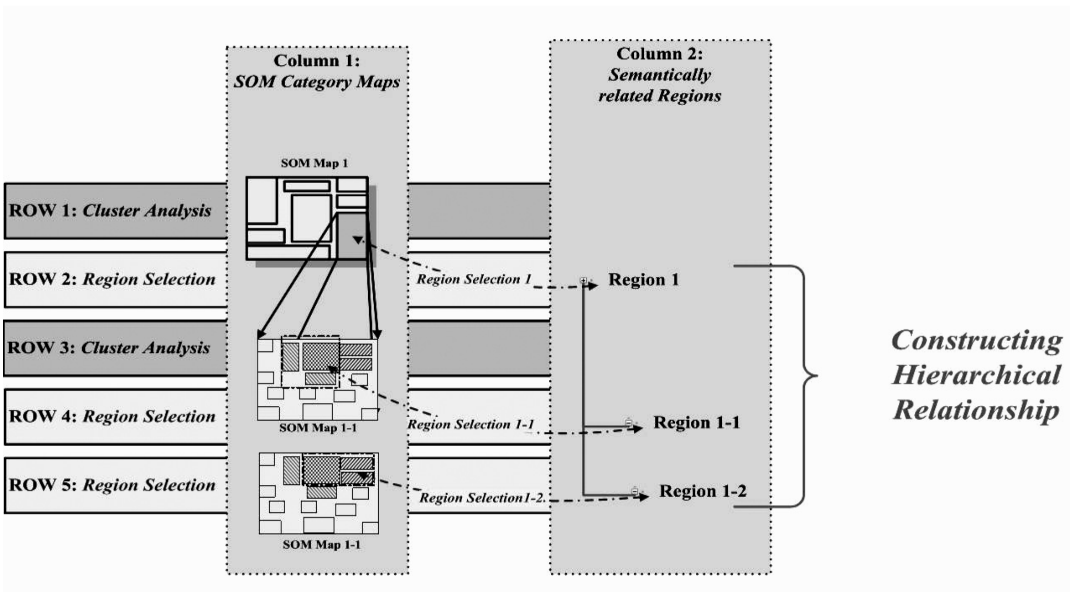
3. Web Service Taxonomy Generation Framework

In this chapter we present our Web service taxonomy generation framework based on the Web service organizing framework, which was discussed in (Hwang, 2010). By extensively observing the contents of the WSDL files, we are able to identify the two categories of semantics that can be inferred from the WSDL file. (1) *Domains of Activity* (what a service deals with) and (2) *Types of Functionality* (what a service does)

We observed that some of the nouns in the list of index terms for a given WSDL file denoted the domains of activity while the verbs indicated the functionality of the service. For example, the nouns *bank, account, route, number, ATM*, and *location* denoted the domain of activity. The verbs *validate, find, retrieve*, and *match* indicated the functionality of the Web service.

3.1 Iterative Clustering Analysis Method

In our Web service taxonomy generation framework, we utilize the SOM clustering algorithm as a core module to semi-automatically generate Web service taxonomies. SOM clustering algorithm generates a two-dimensional map which contains a number of clusters; each cluster may contain a number of semantically related Web services. Within the SOM map, a larger cluster means more Web services are in that classification. Adjacent clusters are more similar in content than nonadjacent clusters.



<Figure 2> Overview of Iterative SOM Cluster Analysis

<Figure 1> illustrates an overview of the iterative SOM cluster analysis method for constructing hierarchical relationships between semantically related regions. In <Figure 1>, each row represents either cluster analysis or semantically related region selection, while each column describes the output of each process in the iterative SOM cluster analysis method. Row 1 and Column 1 in <Figure 1> represent an initial SOM category map (SOM Map 1 in <Figure 1>) with all Web services in our common repository. Once the user generates the initial SOM category map, the user can explore the characteristics of various clusters within the SOM category map by utilizing the Region Selector in our prototype system. The Region Selector allows the user to select semantically related cluster(s) within the SOM category map and save them as semantically related regions. Row 2 and Column 2 illustrate a semantically related region defined by the user (i.e., Region 1 in Figure 2) within the initial SOM category map (SOM Map 1).

After defining the semantically related region, the user re-runs the SOM clustering algorithm to visualize the subset (i.e., the Web services within Region 1) within a new SOM category map (SOM Map 1-1). This process is represented in Row 3 and Column 1 in <Figure 1>. The user then repeats the semantically related region selection process by using the Region Selector within the newly generated SOM category map (SOM Map 1-1). In this example, the user defines two semantically related regions (Region 1-1 and Region 1-2) within the newly generated SOM category map. The user can repeat these steps to define a number of semantically related regions within the newly generated map until no further classification is necessary. By performing such an iterative cluster analysis, the user can construct a hierarchical relationship between the semantically related regions within the parent SOM category map and the semantically related regions within the newly generated child SOM category map.

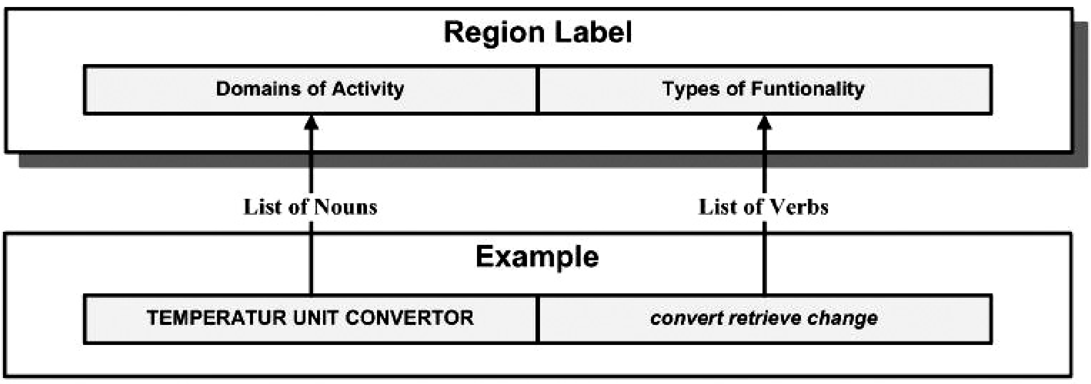
3.2 Descriptive Label Generation Method

To provide users with a more comprehensive Web service taxonomy, the internal Web service taxonomy units should be labeled with some concise names. We found that the labels of subject categories in the existing public Web services repositories are assigned in ambiguous ways; some describe domains of activity (e.g., business and economy) and others describe types of functionality (e.g., validation). Therefore, there is a need to have a consistent way of assigning descriptive labels for subject categories.

Before we present details of our descriptive label generation method for Web service taxonomies, we provide an overview of the prior research on generating labels on cluster analysis. Although it is essential to label clusters, few works have really dealt with this (Muller et al., 1999; Honkela et al., 1997; Lawrie et al., 2001; Glover et al., 2002; Popescul and Ungar, 2000; Merkel and Rauber, 1999). The existing research on generating descriptive labels on clusters can be classified into two approaches : simple term frequency analysis-based approach and statistical analysis-based approach.

The simple term frequency analysis-based approaches rely on the $TF \times IDF$ heuristic for identifying the index terms that best characterize the objects (i.e., documents) mapped on a particular SOM map unit. The work of Honkela et al. perhaps marks the first attempt to generate descriptive labels for the clusters of a SOM map (Honkela et al., 1997). Honkela et al. analyzed the co-occurrence of words in a document and generated a word category map which is further used to represent the various documents contained in the text archive. In (Muller et al., 1999) the cluster labels were chosen as the *n* most frequent terms in the cluster. Merkl and Rauber proposed a straightforward way for assigning labels to the units of a SOM map by utilizing term frequency analysis within the context of a TIME magazine document collection (Merkel and Rauber, 1999). Lawrie et al. extracted salient words and phrases of the instances in a cluster from retrieved documents to organize them hierarchically using a type of co-occurrence known as subsumption (Lawrie et al., 2001).

The statistical analysis-based approaches rely on the statistical analysis for detecting non-descriptive words in hierarchically organized clusters. Glover et al. inferred hierarchical relationships and descriptive labels by employing a statistical model they created to distinguish between the parent, self, and child features in a set of documents (Glover et al., 2002). Popescul and Ungar proposed to use the statistical test χ^2 to detect differences in word distribution across the hierarchy (Popescul and Ungar, 2000). At each cluster node in the hierarchy, starting from the root, the χ^2 test is used to detect a set of words that is equally likely to occur in any sub-cluster of the current node. Those words are considered to be non-descriptive terms and thus are removed from every sub-cluster. After the χ^2 test is used



<Figure 3> Descriptive Label for Semantically Related Region

to remove non-descriptive words from every cluster node, the algorithm labels each cluster with the list of the remaining words at the cluster node ranked by the word frequency.

Our descriptive label generation method relies on both the $TF \times IDF$ heuristic for sorting index terms and the hierarchical relationships among semantically related regions (parent, self, and child) within SOM maps. As shown in Figure 3, each descriptive label for a semantically related region consists of two parts : *domains of activity* and *types of functionality*. The portion of the descriptive label that describes domains of activity is defined with a list of nouns (i.e., represented as upper case letters), while the portion of the descriptive label that describes types of functionality is defined with a list of verbs (i.e., represented as lower case letters).

In order to assign descriptive labels for semantically related regions, we define two descriptive label generation processes based on where the index terms are collected : (1) initial descriptive label generation process and (2) bottom-up fashioned label refining process. The initial descriptive label generation process involves the semantically related region selection process. Once the user defines a semantically related region within a SOM category map, the Region Selector provides the list of index terms that is sorted with the $TF \times IDF$ heuristic. In this process, the index terms are collected from the Web services within the semantically related region. In the sorted list of index terms, more representative index terms appear at the top of the list. By referring to the sorted list of index terms, the user can assign a descriptive label for the newly defined semantically related region.

The bottom-up fashioned label refining process involves the child regions of the semantically related regions. In this process, the index terms are collected from the descriptive labels of the child regions, not from the Web services within the semantically related region. The user then re-assigns the descriptive label for the semantically related region based on the sorted list of index terms. Since the index terms from the descriptive labels of the child regions are more concise and precise than the index terms from the Web services within the semantically related region, the user can assign the descriptive label that reflects the capabilities of the associated child regions.

4. Experimental Design

We now describe our laboratory experiment that evaluates the performance of different Web service search models. Our evaluation objectives were twofold : (1) to evaluate the usability of different Web service search models and (2) to evaluate how the tools (the WebServSOM generated taxonomies and SOM category map) are used to help browsing for Web service discovery.

Forty two human subjects participated in the experiment. Most of the subjects were undergraduate students who are majoring in the Management Information Systems (MIS). Most of them possess fairly good computer skills and

<Table 4> Subjects' Profiles

Attribute	Subjects' Profiles
Gender	27 subjects are male and 15 are female
Education Level	30 subjects are undergraduate students, 4 subjects have earned a bachelor's degree, and 8 subjects did not provide education information
Computer Skill	20 subjects described themselves as expert in their computer skills, 18 subjects described themselves as intermediate in their computer skills, and 4 subjects described themselves as basic in their computer skills.

are familiar with Web search interfaces (keyword searching and tree-based category browsing). No subjects have any prior experience with SOM generated category map browsing and visualization tools. However, a training session was given to all subjects before they conducted search tasks. The training session includes : (1) the introduction of Web services and service-oriented computing, (2) the introduction of the category map and its characteristics, (3) a 15-minute practice session to allow subjects to get familiar with the tools. <Table 4> summarizes the subjects' profiles.

There were three experimental setups of the Web service search strategies : (1) keyword searching (*K*), (2) a combination of keyword searching and tree-based category browsing (*KC*), and (3) a combination of keyword searching, tree-based category browsing, and SOM generated category map browsing (*MKC*). Each experimental setup was implemented with Java Servlet technology and an Oracle 10g Enterprise server. The keyword searching interface consists of an HTML page, which allows users to provide keyword terms and specify the Boolean operators, and the corresponding Java Servlet, which processes the keyword terms from the HTML page, retrieves the Web services from the Oracle 10g Enterprise server, and summarizes the search results into a single HTML page. Tree-based categories for Web services were generated by utilizing the iterative cluster analysis method discussed in Section 4. The labels for the tree-based categories were manually assigned by a human

<Table 5> Summary of Experimental Settings

Search Strategy	Description
Keyword searching (*K*)	Keyword searching interface allows users to search within service name and documentation.
Keyword searching + Tree-based category browsing (*KC*)	Users can search Web services by using either keyword searching or hierarchical category browsing. System allows users to perform keyword searching within a specific category.
SOM generated category map browsing + Keyword searching + Tree-based category browsing (*MKC*)	Users can search Web services by using all three searching interfaces : SOM generated category map browsing, Keyword Searching, and Tree-based Category Browsing.

expert. The tree-based category browsing was incorporated with the keyword searching interface with JavaScript.

In order to incorporate SOM generated category maps into the tree-based category browsing interface, we first created SOM category maps by using an SOM clustering tool. We then saved the SOM category maps as JPEG image files and processed them with image map generation software, which allows us to add hyperlinks within JPEG images. After processing the JPEG image files with the image map generation software, we added them into the tree-based category browsing interface.

During the experiment, each subject used all three setups to perform the searching tasks. The order in which the system was used was randomly assigned to the human subjects to avoid bias due to system sequence. We created 25 sample search tasks based on the 674 WSDL files that were collected from Web services repositories. Each subject was required to perform three search tasks using each system. A time limit of 10 minutes was imposed on each search task. During the experiment, a subject's interaction with the system was recorded as an AVI file. By analyzing the AVI file for each subject, we captured a given subject's detailed activity such as number of clicks, number of search attempts, selection of keywords, and time to accomplish the search task. After finishing the search tasks with a system, a subject rated the system on its usability.

According to (Nielsen and Levy, 1994), measurable usability parameters fall into two broad categories : subjective preference measures and objective performance measures. The subjective preference measures assess how much the users like to use the system, while the objective performance measures indicate how capable the users are at using the system. To measure subjective preference measures, we identified two constructs from the existing usability questionnaires frequently employed in the IS field

<Table 6> Independent Variables and Dependent Variables

Independent Variables	Dependent Variables	
Search Interface : includes K, KC, and MKC	*Objective Performance Measures*	$\text{Precision} = \frac{\text{Number of relevant operations identified by the subject}}{\text{Number of all operations identified by the subject}}$
		$\text{Recall} = \frac{\text{Number of relevant operations identified by the subject}}{\text{Number of relevant operations identified by the expert}}$
		$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$
		Efficiency was measured by the time that participants spent on the search tasks using a search interface.
	Subjective Preference Measures	*Subjective preferences* were measured by two survey constructs : perceived usefulness and perceived ease of use.

(Lewis, 1995; Davis, 1989). The two constructs for the subjective user preference measures consisted of perceived usefulness of the system and perceived ease of use of the system. In this study, there are four objective performance measures : precision, recall, F-measure, and efficiency. They have been widely used in service discovery research studies (Wu and Wu, 2005; Dong et al., 2004; Wang and Stroulia, 2003; Platzer and Dustdar, 2005). The precision measures "how well the search interface helps the users find relevant services by providing less wrong information." The recall measures "how well the search interface helps the users find relevant services by providing more correct information." The efficiency measures "To what extent can the search interface help the user minimize the search task completion time?" <Table 6> summarizes the independent variables and dependent variables in our experiment.

5. Statistical Analysis and Experimental Results

After obtaining performance measures, we used the ANOVA and Tukey's post hoc analyses to test whether the performance measures of different experimental settings were significantly different. To evaluate the effectiveness of three different experimental settings, we used the four objective performance measures (i.e., precision, recall, F-measure, and efficiency).

<Table 7> summarizes the ANOVA test results, showing significant differences in all four objective performance measures across different experimental settings. For instance, the F-ratio of precision is 0.387 and *p*-value is 0.000. This means that the difference between the precision of the three different experimental settings is significant at the level of 0.000. Since there is a significant difference among the different ex-

<Table 7> ANOVA Test Results of Objective Performance Measurements

		Sum of Squares	df	Mean Square	F	Sig.
Precision	Between Groups	0.775	2	0.387	25.210	0.000
	Within Groups	1.890	123	0.015		
	Total	2.665	125			
Recall	Between Groups	0.024	2	0.012	8.333	0.000
	Within Groups	0.177	123	0.001		
	Total	0.201	125			
F-measure	Between Groups	0.527	2	0.264	27.500	0.000
	Within Groups	1.179	123	0.010		
	Total	1.706	125			
Efficiency	Between Groups	31774.737	2	15887.369	10.936	0.000
	Within Groups	178696.762	123	1452.819		
	Total	210471.499	125			

perimental settings, we conducted Tukey's post hoc test to determine exactly which means are significantly different from which other ones.

<Table 8> lists the Tukey's post hoc test results with the three effectiveness measures. In terms of precision, the Tukey's post hoc test grouped the three different experimental settings into two homogenous subsets in the following order : *MKC, KC* > *K*. In terms of recall, the Tukey's post hoc test classified the three different experimental settings into two homogenous subsets in the following order : *MKC, KC* > *K*. In terms of F-measure, the Tukey's post hoc test organized the three different experimental settings into two homogenous subsets in the following order : *MKC, KC* > *K*. Based on these findings, we concluded that there was no significant difference between the combined keyword searching and tree-based category browsing (*KC*), and the combined keyword searching, tree-based category browsing, and SOM generated category map browsing (*MKC*). However the combined multiple search strategies (*MKC* and *KC*) outperformed the simple keyword searching (*K*) across all three effectiveness measures. In terms of efficiency (i.e., time requirements), the Tukey's post hoc test grouped the three different experimental settings into two homogeneous subsets in the following order : *MKC, KC* > *K*. This finding confirms that using SOM generated category map browsing did not significantly increase the time required for Web service searching.

To evaluate the usability of the three Web

<Table 8> Summary of Tukey's Post Hoc Tests with Objective Performance Measures

		N	Subset for alpha = .05	
Measure	Index		1	2
Precision	*K*	42	0.592	
	KC	42		0.771
	MKC	42		0.799
	Sig.		1.000	0.954
Recall	*MKC*	42	0.429	
	KC	42	0.387	
	K	42		0.154
	Sig.		0.881	1.000
F-measure	*MKC*	42	0.515	
	KC	42	0.497	
	K	42		0.258
	Sig.		0.981	1.000
Efficiency	*K*	42	124.111	
	KC	42		150.254
	MKC	42		162.127
	Sig.		1.000	.327

<Table 9> Descriptive Statistics of Subjective Preference Measures

[a] The range of rating is from 1 to 7 with 1 being the best.

		N	Mean	Std. Deviation	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Usefulness	*K*	38	4.733[a]	1.347	4.290	5.175	2.330	7.000
	KC	38	1.737[a]	0.594	1.542	1.932	1.000	4.000
	MKC	38	1.623[a]	0.5380	1.446	1.800	1.000	2.830
	Total	114	2.697	1.702	2.382	3.013	1.000	7.000
Ease of Use	*K*	38	1.904[a]	1.013	1.571	2.237	1.000	6.000
	KC	38	1.776[a]	1.366	1.327	2.225	1.000	6.500
	MKC	38	1.820[a]	0.843	1.543	2.097	1.000	6.000
	Total	114	1.833	1.088	1.632	2.035	1.000	6.500

service search strategies, we used two performance measures : perceived usefulness and perceived ease of use. <Table 9> shows the average rating on the perceived usefulness and the perceived ease of use for each setting. We conducted the ANOVA test for the two performance measures to see if there is any significant difference between different experimental settings.

<Table 10> summarizes the ANOVA test results, showing significant difference in the perceived usefulness across three different experimental settings. It also shows that there is no significant difference in the perceived ease of use. Based on the result of the ANOVA test for the perceived ease of use, we conclude that all three Web service search strategies achieved the same level of perceived ease of use. Since there is a significant difference in the perceived usefulness among the different experimental settings, we conducted Tukey's post hoc analysis with the perceived usefulness to determine exactly which means are significantly different from which other ones.

<Table 11> lists the Tukey's post hoc analysis result based on the perceived usefulness of the three different experimental settings. The

<Table 10> ANOVA Test Results of Subjective Preference Measures

		Sum of Squares	df	Mean Square	F	Sig.
Usefulness	Between Groups	236.317	2	118.159	144.317	.000
	Within Groups	90.881	111	0.819		
Ease of Use	Between Groups	0.320	2	0.160	0.133	.875
	Within Groups	133.328	111	1.201		

Tukey's post hoc test grouped the three different experimental settings into two homogeneous subsets in the following order : *KC, MKC > K*. The result of the Tukey's post hoc analysis reveals that the participating human subjects' perception was that the combined multiple search strategies (i.e., *KC* and *MKC*) were more useful than the keyword searching (*K*). Based on the results of our experiment with three different Web service search strategies, we found the following :

<Table 11> Tukey's Post Hoc Test Result with Perceived Usefulness

Index	N	Subset for alpha = .05	
		1	2
MKC	38	1.623	
KC	38	1.737	
K	38		4.733
Sig.		.847	1.000

- The combined multiple Web service search strategies (*MKC* and *KC*) outperformed the keyword searching (*K*) in terms of objective performance measures (precision, recall, F-measure and time cost).
- The SOM generated category map browsing did not significantly increase the time required for Web service searching.
- The participating human subjects' perception was that the combined multiple search strategies (*MKC* and *KC*) were more useful than keyword search (K).
- There was no significant difference in the perceived ease of use across the three different Web service search strategies.

6. 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 Web services search models actually exist and what search model is required for effective Web service discovery.

In this article we presented an empirical evaluation that compares the usability of different search models in Web service repositories. In our evaluation methodology, we compared usability of different search models in terms of objective performance measures and subjective performance measures. Our findings suggest that the combined multiple Web service search models outperformed the simple keyword searching while the SOM generated category map browsing did not significantly increase the time required for Web service searching. In addition, the participating human subjects expressed that the combined multiple search models were more useful than simple keyword search model. In closing there is a gap between the traditional Web service search models, which are employed in most exiting Web service repositories, and a desirable Web service search model. Consequently, the findings from this study reveal that our combined multiple search model fills the gap between them.

References

Basili, V. R., L. C. Briand, and M. W. L, "How Reuse Influences Productivity in Object-Oriented Systems", *Communications of the ACM*, Vol.39, No.10(2003), 104～116.

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.

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

Chávez, A., C. Tornabene, and G. Wiederhold, "Software Component Licensing Issues : A Primer", *IEEE Software*, Vol.15, No.5 (2000), 47～52.

Chen, H., R. R. Robin and B. R. Schatz, "Internet Browsing and Searching : User Evaluations of Category Map and Concept Space Techniques", *Journal of the American society for information science*, Vol.49, No.7(1998), 582～603.

Davis, F. D., "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*, Vol. 13, No.3(1989), 319～340.

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

Glover, E., D. M. Pennock, S. Lawrence and R. Krovetz, "Inferring Hierarchical Descriptions", in *Proceedings of the 11th International Conference on Information and Knowledge Management CIKM'02*, McLean, VA, (2002).

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. Kushmerick, "Semi-atuomatically Annotating Semantic Web services", in *Proceedings of Semantic Web Conference*, (2004).

Honkela, T., S. Kaski, K. Lagus and T. Kohonen, "WEBSOM-self-organizing maps of document collections", in *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, Espoo, Finland, (1997).

Hwang, Y., "Facilitating Web Service Taxonomy Generation : An Artificial Neural Network based Framework, A Prototype Systems, and Evaluation", *Journal of Intelligence and Information Systems*, Vol.16, No.2(2010), 33~52.

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

Kokash, N., W. Heuvel and V. D'Andrea, "Leveraging the Web service Discovery with Customizable Hybrid Matching", In *Technical Report DIT-06-042*, University of Trento, (2006).

Krueger, C. W., "Software reuse", *ACM Computing Surveys*, Vol.24, No.2(1992), 131～183.

Lawrie, D., W. B. Croft and A. Rosenberg. "Finding topic words for hierarchical summarization", in *Proceedings of the 24th annual international ACM SIGIR conference on Research and Development information Re-*

trieval, New Orleans, Louisiana, United States, (2001).

Lewis, J. R., "IBM computer usability satisfaction questionnaires : Psychometric evaluations and instructions for use", *International Journal of Human-Computer Interaction,* Vol.7, No.1 (1995), 57～78.

Manber, U., M. Smith and B. Gopal, "Web-Glimpse-Combining Browsing and Searching", in *Proceedings of the USENIX 1997 Annual Technical Conference,* Anaheim, CA. (1997).

Marchionini, G., "An invitation to browse : Designing full text systems for novice users", *Canadian Journal of Information Science,* Vol.12, No.3(1987), 69～79.

Marchionini, G. and B. Shneiderman, "Finding facts vs. browsing knowledge in hypertext systems", *IEEE Computer,* Vol.21, No.3 (1988), 70～79.

Merkl, D. and A. Rauber. "Automatic labeling of self-organizing maps for information retrieval", in *Proceedings of the 6th International Conference on Neural Information Processing ICONIP'99,* Perth, WA, Australia, (1999).

Muller, A., J. Dorre, P. Gerstl and R. Seiffert, "The TaxGen Framework : Automating the Generation of a taxonomy for a large document collection", in *Proceedings of the 32nd Hawaii International Conference on System Science (HICSS), Maui, Hawaii),* pp. 20~34, (1999).

Nielsen, J. and J. Levy, "Measuring usability : preference vs. performance", *Communications of the ACM,* Vol.37, No.4(1994), 66～75.

Pahlevi, S. M. and H. Kitagawa, "Taxonomy-based context conveyance for Web search", *LECTURE NOTES IN COMPUTER SCIENCE,* Vol.28, No.88(2003), 536～553.

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

Poulin, J. S., J. M. Caruso and D. R. Hancock, "The business case for software reuse", *IBM Systems Journal*, Vol.32, No.4(1993), 567～594.

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).

Popescul, A. and L. Ungar, "Automatic labeling of document clusters", Unpublished manuscript http : //citeseer.nj.nec.com/popescul00automatic.html, (2000).

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 J. Koehler, "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.

Zhuge, H. and J. Liu, "Flexible Retrieval of Web Services", *The Journal of Systems and Software*, Vol.70, No.2(2004), 107～116.

Abstract

웹서비스 저장소의 검색기법에 관한 실증적 연구

황유섭*

월드와이드웹 (WWW)은 유용한 정보를 포함하는 자료들의 집합에서 유용한 작업을 수행할 수 있는 서비스들의 집합으로 변화하고 있다. 새롭게 등장하고 있는 웹서비스 기술은 향후 웹의 기술적 변화를 추구하며 최근 웹의 변화에 중요한 역할을 수행할 것으로 기대된다. 웹서비스는 어플리케이션 간의 통신을 위한 호환성 표준을 제시하며 기업 내/외를 아우를 수 있는 어플리케이션 상호작용 및 통합을 촉진한다. 웹서비스가 서비스 지향 컴퓨팅환경으로서 운영하기 위해서는 웹서비스 저장소가 완성도 높게 구축되어 있어야 할 뿐 아니라, 사용자들의 필요에 맞는 웹서비스 컴포넌트를 찾을 수 있는 효율적인 도구들을 제공하여야 한다. 서비스 지향 컴퓨팅을 위한 웹서비스의 중요성이 증대됨에 따라 웹서비스의 발견을 효율적으로 지원할 수 있는 기법의 수요 또한 증대된다. 다수의 웹서비스 저장소들은 웹서비스 분류체계 및 검색기법들을 제안하여 왔지만, 대부분의 분류체계와 기존의 검색기법들은 실질적으로 활용하기에는 제대로 발달하지 못하였거나 지속적이고 체계적으로 관리하기에 너무 어려운 단점을 갖고 있다.

이 논문에서는 인공신경망 기반 군집화 기법과 XML 기반의 웹서비스 기술표준인 WSDL의 의미적 가치를 활용하여 웹서비스 분류체계 생성 프레임워크를 통한 복합 검색기법을 제안한다. 이 논문에서 인공신경망을 활용하여 제안하는 웹서비스 분류체계 생성 프레임워크는 실증적인 프로토타입 시스템으로 개발하였으며, 실제 운영되고 있는 웹서비스 저장소로부터 획득한 실제 웹서비스들을 사용하여 제안하는 웹서비스 복합 검색기법을 실증적으로 평가하였다. 또한 제안하는 방식의 효용성을 보여주는 의미 있는 실험결과를 보고한다.

Keywords : 웹서비스, 시맨틱, 검색기법, 웹서비스 발견, 실증연구, 설문, 시스템 유용성, 인공신경망

* 서울시립대학교 경영대학

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 관심분야는 service-oriented computing, forecasting, artificial neural network의 활용 방안 연구, IT strategy 등이다.