Facilitating Web Service Taxonomy Generation : An Artificial Neural Network based Framework, A Prototype Systems, and Evaluation

Yousub Hwang
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 both within and across enterprises. To make Web services for service-oriented computing operational, it is important that Web service repositories not only be well-structured but also provide efficient tools for developers to find reusable Web service components that meet their needs. As the potential of Web services for service-oriented computing is being widely recognized, the demand for effective Web service discovery mechanisms is concomitantly growing. A number of public Web service repositories have been proposed, but the Web service taxonomy generation has not been satisfactorily addressed. Unfortunately, most existing Web service taxonomies are either too rudimentary to be useful or too hard to be maintained.

In this paper, we propose a Web service taxonomy generation framework that combines an artificial neural network based clustering techniques with descriptive label generating and leverages the semantics of the XML-based service specification in WSDL documents. We believe that this is one of the first attempts at applying data mining techniques in the Web service discovery domain. We have developed a prototype system based on the proposed framework using an unsupervised artificial neural network and empirically evaluated the proposed approach and tool using real Web service descriptions drawn from operational Web service repositories. We report on some preliminary results demonstrating the efficacy of the proposed approach.

Received : May 22, 2010 Revision : June 04, 2010 Accepted : June 14, 2010 Corresponding author : Yousub Hwang

1. Introduction

One of the industry standards for Web service repositories is Universal Description, Discovery, and Integration (UDDI). The search mechanisms provided by most UDDI registries are based on either keyword search or browsing through predefined hierarchical business categories. Keyword search is known to have the drawback that users may irrelevant search results due to such issues as synonyms and word ambiguity. Consequently, users may have to spend a lot of time browsing through the search results to identify the Web services that most closely meet their needs.

If the Web services are categorized well, browsing through predefined categories may be a better alternative for service discovery. However, it requires both service providers and consumers to have prior knowledge of the service categorization schemes, such as the North American Industry Classification System (NAICS) and the United Nations Standard Products and Services Code (UNSPSC). In other words, the service providers must publish their Web services in the appropriate UDDI business categories and the service consumers must browse the 'right' business categories to find the potentially relevant Web services. Thus, although UDDI provides a standard interface for simple keyword based search and predefined business category based browsing, more effective mechanisms are still needed to improve the current search mechanisms for discovering semantically relevant Web services.

The main objective of this research is to develop a more effective mechanism for Web service taxonomy generation framework. We propose a Web service taxonomy generation framework that combines clustering techniques with descriptive label generating and leverage the semantics of the XML-based service specification in WSDL (Web Service Description Language) documents. We have developed a software tool for Web service taxonomy generation 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 service repositories. We report on some preliminary results demonstrating the efficacy of the proposed approach. The remainder of this paper is organized as follows. We begin in Section 2 with a overview of public Web service repositories to illustrate limitations of current Web service taxonomies in public Web service repositories. In Section 3, we discuss the semi-autonomous Web service taxonomy generation methods implemented in the prototype. We then report on our empirical evaluation in Section 4. Finally, we conclude the paper in Section 5.

2. 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., *BindingPoint*, *NetXML*, *and SalCentral*) and adding newly available repositories (i.e., *RemoteMethods and Woogle* [Dong et al., 2004]). 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.

Universal Description, *Discovery*, *and Integration* (*UDDI*) is one of the industry standards for Web services repositories. UDDI is jointly proposed by IBM, Microsoft and Ariba. It provides service registry architecture for businesses to build a registry, discover each other, and learn how to interact over the Internet. UDDI supports both query-based search and taxonomy browsing. The drawback of UDDI's query-based searching is that many irrelevant search results may be returned due to issues such as substring matching. For instance, when searching for "date", any services that contain words such as "validate" or "update" (which are clearly not related to date) are returned. Furthermore, UDDI doesn't support any advanced search options for further refining query-based search.

Browsing can be done according to industry/product standard classification schemes based on the North American Industry Classifiction System (NAICS) and the United Nations Standard Products and Services Code (UNSPSC). For example, the UNSPSC-based product classification is a hierarchical classification of products and services with five levels. The first level of the UNSPSC product classification contains more than 400 general product types. Thus, service consumers need to explore a large product standard classification scheme in order to browse appropriate Web services. This industry/product standard classification-based browsing is clearly insufficient, because it relies on the shared common-sense understanding of the application domain by the users who publish and consume the specified services [Stroulia and Wang, 2005]. In other words, the service providers must publish their Web services in the appropriate UDDI industry/product categories, and the service consumers must browse the 'right' industry/product categories to find potentially relevant Web services. In addition, there is no guarantee that the industry/product standard classification actually represents the underlying functionality of a service and not something else. Although UDDI supports both query-based search and taxonomy browsing, it does not allow the user to further refine a query-based search by restricting it to a given category within the industry/product standard classification schemes. UDDI provides search results with a brief summary of each service, including the following: service name, a textual description, and a hyperlink to the WSDL file with which it is associated.

WebServiceX is a Web service provider that currently offers about 168 Web services.

These services are grouped into seven categories which form the basic browsing mechanism. Each category contains a number of Web services which ranges from 1 to 19. These categoriesare ambiguously created by using multiple category schemes. For example, some categories denote the domain of Web services (e.g., Communications, Business/Commerce, and Graphics) while others name a certain functionality type (e.g., Conversion, Lookup, and Value manipulation). *WebServiceX* doesn't provide any query-based search facilities. Each service listed in the browsing results is displayed with the service name, a textual description, and a hyperlink to the WSDL file with which it is associated.

WebServiceList provides 17 categories for browsing the available services (an estimated 480). Each category describes a number of Web services ranged from 1 to 176. Like the taxonomy in *WebServiceX*, these categories are ambiguously populated by using multiple category schemes (e.g., either the domain of Web services or a type of functionalities). Further, there is a mismatch between the content provided by the Web services to be categorized and that covered by the categories. For example, the retail service category contains currency convert and health care provider search Web services, which are supposed to be categorized into the conversion services and the healthcare services category respectively. In addition, Web services can be browsed alphabetically, but this function is not very helpful when the individual does not know the exact name of the service for which he/she is looking. This repository offers a query-based search that matches the name and description of Web services. Unlike the UDDI search mechanism, this search works on correct tokenization (i.e., it matches search terms to whole words only and not to substrings in the given text). However, this repository doesn't offer any advanced search options. Although this repository supports both query-based search and taxonomy browsing, it doesn't allow users to further refine a query-based search by restricting it to a given subject category. Each service listed in the search results is represented with the service name, a textual description, a hyperlink to the WSDL file with which it is associated, and the service rating score.

Xmethods is one of the largest Web services repositories, containing more than 500 Web services. However, this site only provides a long list of services. It has no support for browsing, nor does it provide any search facilities. Each service in the list is summarized with service name, a textual description, and a hyperlink to the WSDL file with which it is associated.

RemoteMethods has more than 300 registriesof Web services. This site offers both search and browse facilities simultaneously. Searching for a keyword will return any Web service that contains the keyword as a substring of the strings denoting the Web service's name and description. This repository doesn't support any advanced search options. Browsing the available Web services can be done via eight top categories, which are further specialized into two levels (45 categories total). Like the taxonomy in *WebServiceX*, these categories are ambiguously populated using multiple category schemes (e.g., either the domain of Web services or a type of functionalities). Each category contains a number of Web services ranged from 1 to 119. This repository provides search results using the following: the service name, a textual description, the service price, a hyperlink to the WSDL file with which it is associated, the service rating, the service reviews, and the number of hits.

Woogle is a research project aimed at supporting automated service discovery. Like *RemoteMethods*, this repository provides both search and browsing facilities simultaneously. *Woogle* allows the user to further refine a query-based search by restricting it to a given subject category. *Woogle* supports twotypes of searches : (i) a keyword search based on service names and textual descriptions and (ii) a template search on operation, which allows for a query-based search based on operation names and documentation. *Woogle* is the only repository that offers advanced search options, such as Booleanoperator with multiple keyword terms. It returns search results with a brief summary of services that include the following : the service name, a textual description, the status, and a hyperlink to the WSDL file with which it is associated. This repository supports browsing the availableWeb services (estimated to be 600) through 42 hierarchically organized categories (8 categories at the first level and these categories are further specialized into three levels). Each category describes a number of Web services ranged from 1 to 96. Like the taxonomy in *WebServiceX*, these categories are ambiguously populated using multiple category schemes (e.g., either the domain of Web services or a type of functionalities).

As listed in <Table 1>, we encountered three types of Web service taxonomies : alphabetically organized 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

<Table 1> Summary of Taxonomy Browsing in Public Web Services Repositories

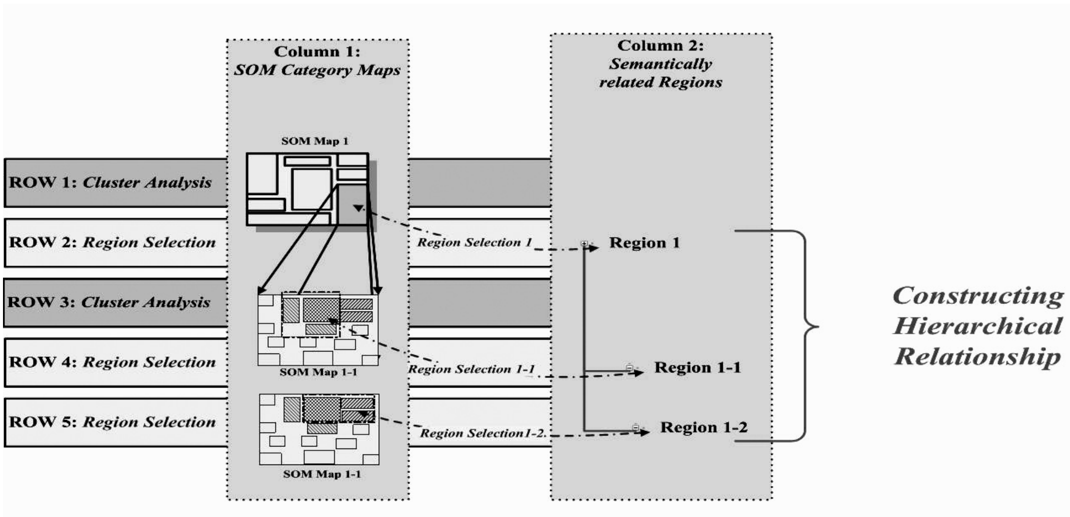
Repository	Taxonomy Browsing			
	Type of taxonomy	Scope of taxonomy	Size of taxonomy	Depth of taxonomy
UDDI	Standard classification schemes	Industry/product sector	16,000	5
WebServiceX	Lightweight	ambiguous	7	1
WebServiceList	Alphabetically Organized	ambiguous	17	1
Xmethods	None	None	None	None
RemoteMethods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

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 overpopulated 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 ofactivity (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 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. 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 crea-



<Figure 1> Overview of Iterative SOM Cluster Analysis

ted and populated with instances.

3. Semi-autonomous Web Service Taxonomy Generation Method

3.1 Iterative SOM Cluster Analysis

In our Web service taxonomy generation framework, we utilize the SOM clustering algorithm (Kohonen, 2001) as a core module to semi-automatically generate Web service taxonomies. SOM clustering algorithm generates a two-dimensional map which contain 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 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 1>) within the initial SOM category map (SOM Map 1).

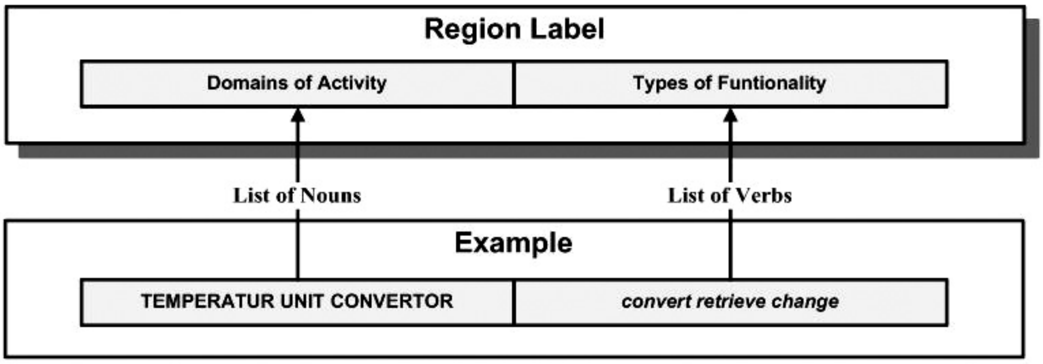
After defining the semantically related region, the user re-runs the SOM clustering algorithm to visualize the subset (i.e., the Web services within Region 1) within a new SOM category map (SOM Map 1-1). This process is represented in Row 3 and Column 1 in Figure 1. The user then repeats the semantically 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 categoriesin 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; Merkl 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 descriptivelabels for the clusters of a SOM map (Honkela et al., 1997). Honkela et al. analyzed the co-occurrenceof 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. the cluster labels were chosen as the n most frequent terms in the cluster (Muller et al., 1999). Merkl and Rauber proposed a straightforwardway for assigning labels to the units of a SOM map by utilizing term frequency analysis within the context of a TIME magazine document collection (Merkl and Rauber, 1999). Lawrie et al. extracted salient words and phrases of the instances in a



<Figure 2> Descriptive Label for Semantically Related Region

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 analysisfor 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 et al. proposed to use the statistical test x^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 x^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 x^2 test is used 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.

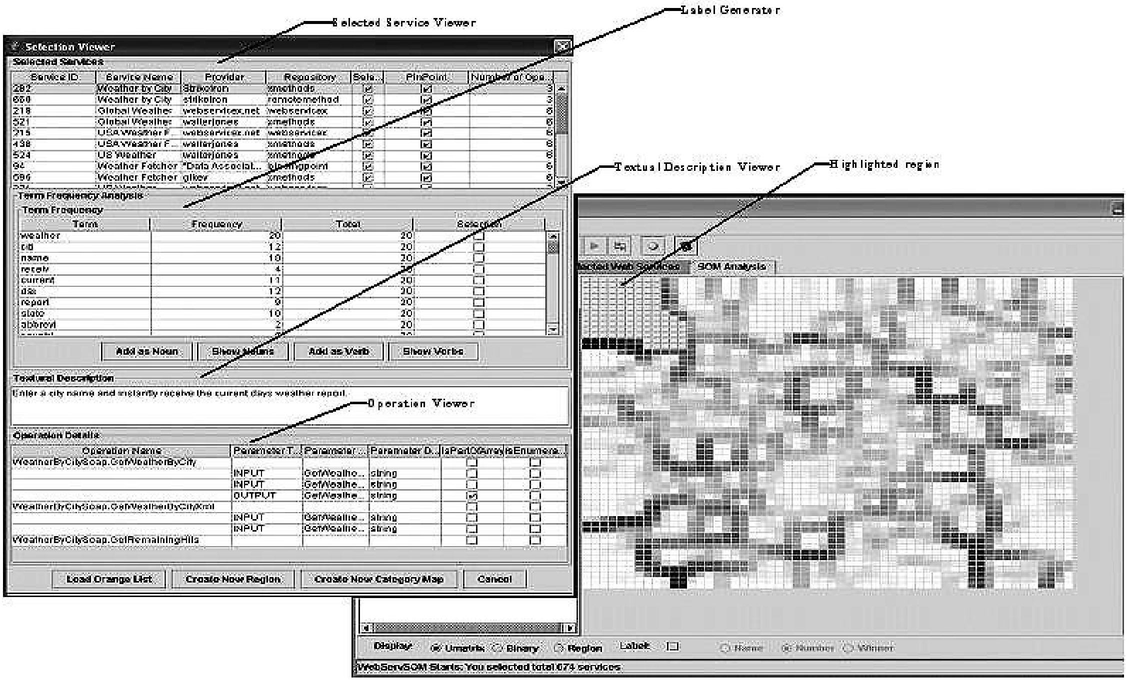
Our descriptive label generation method relies on both the $TF \times IDF$ heuristic for sorting index terms and the hierarchical relationships among semantically related regions (parent, self, and child) within SOM maps. As shown in <Figure 2>, each descriptive label for a semantically related region 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).

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 precisethan 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.

3.3 Prototype System

In this section, we present the publisher interface in our prototype system as a realization of the proposed Web service taxonomy generation framework. <Figure 3> illustrates the clustering result from the initial SOM cluster analysis of the 674 WSDL files. According to our previous work (Hwang, 2009), SOM is a promising clustering algorithm for semantically or-



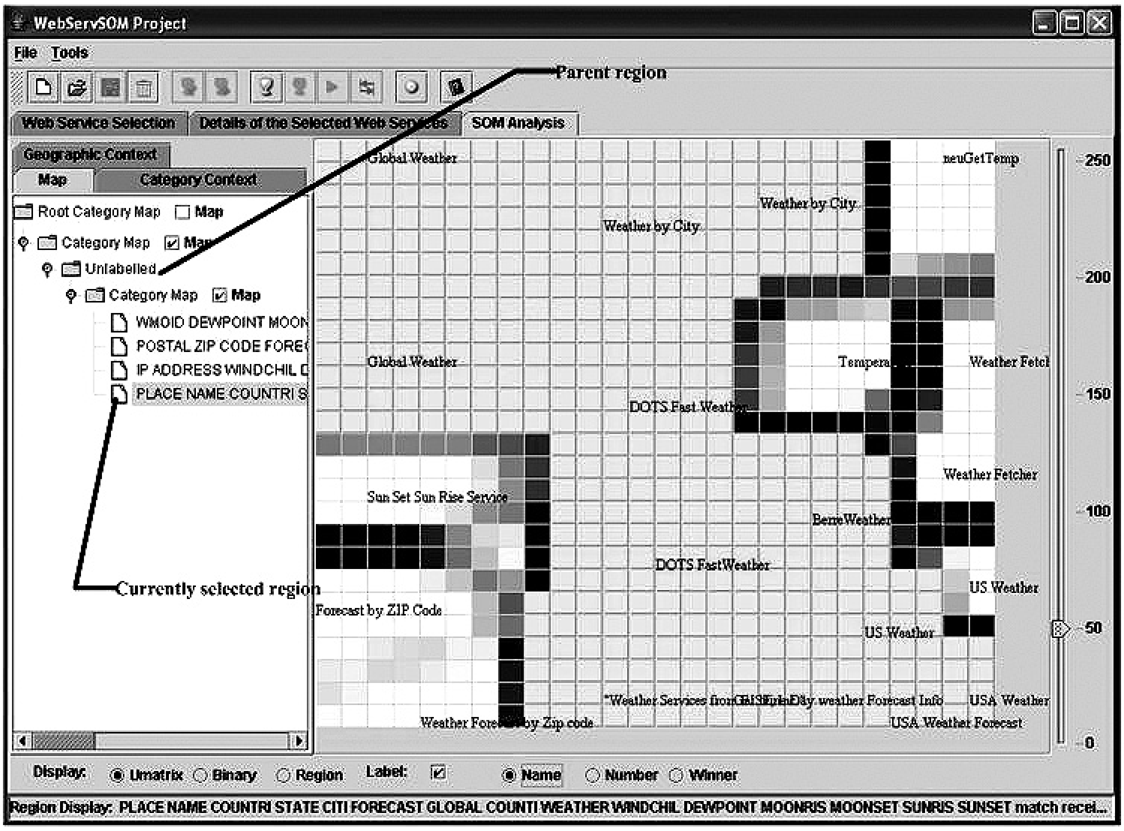
<Figure 3> Clustering Result from the SOM Cluster Analysis

ganizing Web services. The user can select semantically related cluster (s) within the SOM category map by utilizing the Region Selector in the prototype system. The Region Selector supports the following user actions : (1) explore the characteristics of various clusters within the SOM category map by displaying the detailed contents of the WSDL files, (2) define semantically related regions within the SOM category map, and (3) assign a descriptive label for a given semantically related region.

As illustrated in <Figure 3>, the Region Selector dialog interface consists of four interface components : Selected Service Viewer, Label Generator, Textual Description Viewer, and Operation Viewer. The Selected Service Viewer displays the currently selected Web services in the SOM category map in table format. The Label Generator displays the sorted index terms based on the $TF \times IDF$ heuristic and allows the user to define a descriptive label for the currently selected regions within the SOM category map. More representative index terms appear at the top of the sorted list of index terms. The user may either assign a descriptive label for the selected region by referring to the list of index terms, or leave the selected region as unlabelled. The unlabelled regions will be renamed after completing further cluster analysis and a semantically related sub-regions identification process.

After defining the semantically related region within the parent SOM category map, the user can conduct further cluster analysis with the selected Web services within the parent SOM category map. In this example, the 20 Web services that retrieve weather forecast information have been selected within the semantically related region. <Figure 4> illustrates the newly generated SOM category map with the 20 weather forecast Web services. The newly generated SOM category map initially indicates that there are seven duplicates by highlighting clusters with red-colored lines (see <Figure 4>). By utilizing the Region Selector, users can identify four semantically related regions within the newly created map, depending on the four different types of inputs for retrieving weather forecast information : (1) by place name (i.e., names of city, county, state, or country), (2) by Internet Protocol (IP) address, (3) by postal zip code, and (4) by World Meteorological Organization (WMO) identification number. Web services can support multiple types of inputs for retrieving weather forecast information. For instance, the *DOTS Fast Weather* Web service has the ability to retrieve weather forecasting information based on all four types of inputs. For this reason, the cluster that contains the *DOTS Fast Weather* Web services is located in the center of the newly created map (refer to <Figure 4>).

Based on the hierarchical relationships between the semantically related regions, the Taxonomy Extractor derives a domain-independent Web service taxonomy that reflects domains of activity and types of functionality. Consequently, the Taxonomy Extractor generates a hierarchicallyorganized 71-Web service subject context taxonomy : the top level of the taxonomy



<Figure 4> Further Cluster Analysis with the Selected Services

consists of 17 subject categories, each of which contains from 0 to 17 subcategories, with an average of 3.579 and sample standard deviation of 4.299.

4. Evaluation

In order to compare the Web service taxonomies of both *BindingPoint.com* and *WebServiceX.net* with the WebServSOM generated service taxonomies, we conducted a set of experiments. Our experiments consisted of five different steps sample data collection, input data generation, Web service taxonomy generation, taxonomy evaluation, and statistical analysis. We explain our experimental design in detail according to these five steps.

4.1 Experimental Design

We collected sample data from both *BindingPoint.com* and *WebServiceX.net* because both of these repositories organize Web services into hierarchical browsing interfaces that help service consumers locate Web services. We obtained a total of 357 WSDL files, 189 from *BindingPoint.*

com and 168 from *WebServiceX.net.* In the data collection process, we collected service name, service provider information, textual description, and WSDL file, as well as Web service taxonomy information from the corresponding Web services repository.

The two Web services repositories categorize Web services differently. *BindingPoint.com* has classified 189 Web services into 36 different categories. The Web service taxonomy in *BindingPoint.com* is hierarchically organized : the first level consists of 7 general subject categories, each of which contains from 0 to 8 sub-categories, with an average of 3.429 and standard deviation of 2.636. There are 24 subject categories in the second level of the Web service taxonomy in *BindingPoint.com*. Only four of the 24 subject categoriesin the second level contain from 1 to 2 sub-categories, with an average of 1.25 and standard deviation of 0.5. *WebServiceX.net* has classified 168 Web services into 7 general subject categories, which were not further specialized. Because *BindingPoint.com* provides more specially categorized Web services than *WebServiceX.net*, we might expect the following : (1) a Web service taxonomy of *BindingPoint.com* results in greater accuracy than one of *WebServiceX.net* (2) the Web service taxonomy of *WebServiceX.net* has a low false negative rate (purity across-subject categories) compared to *BindingPoint.com.*

After completing the sample data collection stage, we obtained two sample data sets from two Web services repositories : one sample data set consisted of 189 Web services from *BindingPoint.com* and the other sample data set contained 168 Web services from *WebServiceX.net.* Based on each sample data set, we randomly sampled input data for the further experimental stages. Each input data set consisted of 30 services. Wecreated 100 input data sets from each sample data set and continued further experimental stages with these data sets.

In the Web service taxonomy generation stage, we generate three Web service taxonomies for each input data set : the WebServSOM generatedWeb service taxonomy, the original taxonomy gathered from the corresponding Web services repository, and the human experts generated Web service taxonomy model solution.

In order to compare the quality of the original taxonomy information from the corresponding Web services repository for a given input data set and the WebServSOM generated taxonomy, human experts initiallyexamined each randomly sampled input data set and categorized them into small groups. Four human experts were recruited to generate a Web service taxonomy for each input data set. The recruited human experts were business students (two graduates and two undergraduates) who are majoring in Management Information Systems. These human experts also had prior application development experience and understanding of Web service technology. In our experiments, the human generated taxonomy model for each input data set was considered the decision base to evaluate the Web service taxonomies that were generated by the prototype system and collected from the corresponding Web

<Table 2> Characteristics of the Human Experts Generated Service Taxonomies

Service Registry	Number of subject categories		Number of *multiple-member* categories		Number of *single-member* categories	
	Mean	STDV	Mean	STDV	Mean	STDV
BindingPoint.com	23.667	2.368	4.867	1.407	18.8	3.438
WebServiceX.net	17.567	2.207	4.767	1.381	12.8	2.721
WebServSOM	20.367	2.332	4.776	1.349	16.3	3.832

services repository.

We examined general characteristics of the human experts generated service taxonomy information for each input data set as listed in <Table 2>. The input data sets from *BindingPoint.com* were grouped into 20 to 29 subject categories, with an average of 23.667 and a sample standard deviation of 2.368, while the input data sets from *WebServiceX.net* were grouped into 13 to 21 subject categories, with an average of 17.567 and a sample standard deviation of 2.207. The input data sets from *BindingPoint.com* contain a significantly higher average number of single-member categories than the input data sets from *WebServiceX.net*, while the input data sets from both *BindingPoint.com* and *WebServiceX.net* have a similaraverage number of multiple-member categories (4.867 and 4.767 respectively). Because the input data sets from *BindingPoint.com* were randomly sampled from a much larger sample, they contain a relatively higher number of single-member categories than do the input data sets from *WebServiceX.net*.

To compare the taxonomy performance of WebServSOM with both *BindingPoint.com* and *WebServiceX.net*, we applied the contingency table model-based evaluation framework. To score

<Table 3> Contingency Table Model for Evaluation Measurement

System Answer	Human Expert Answer	
	Yes	No
Yes	a	b
No	c	d

our results, both the system generated Web service taxonomies and the human experts generated service taxonomy model solutions were converted into two lists of yes-no answers to the co-occurrence question, “Does the pair of services belong to the same subject category?” for each pair of services (Swets, 1969; Lewis, 1991; Yang, 1999). Let n be the number of services. Then there are $\frac{n(n-1)}{2}$ pairs of services. Each pair must fall into one of four categories. If A, B, C, and D are the number of service pairs in each case, then $A+B+C+D=\frac{n(n-1)}{2}$ (as in <Table 3>).

In general, a contingency table is constructed by counting the number of observed associations in the co-occurrence answer lists. Since the human experts’ judgments did not always agree, we used fractional values for the correctness of each answer instead of binary classifiers (0 for "incorrect" and 1 for "correct") (Hatzivassiloglou and McKeown, 1993). We defined the correctness of each answer as the relative frequency of the cluster assignments between the two Web services among the human experts generated servicetaxonomy models; in this way, subject category assignments receive a correctness value proportional to their popularity among the human judges.

Once correctness values have been defined, we can generalize measures such as "the number of correct subject category assignments generated by a given experimental setting" using a summation of these values instead of counting. The subject category assignments can be evaluated using a two-way contingency table (See <Table 3>) for each subject category, which has four cells where

- cell a sums up the correctness values for the Web services correctly assigned to this subject category;
- cell b sums up the correctness values for the Web services incorrectly assigned to this subject category;
- cell c sums up the correctness values for the Web services incorrectly rejected from this subject category;
- cell d sums up the correctness values for the Web services correctly rejected from this subject category.

Based on the contingency table model (Refer to <Table 3>), the following performance measures are derived :

- Accuracy (Acc) = $\dfrac{A+D}{A+B+C+D}$
- False Positive (fp) = $\dfrac{B}{A+B}$
- False Negative (fn) = $\dfrac{C}{A+C}$

We can estimate overall quality of a Web service taxonomy by referring to accuracy. Since the false positive rate is the proportion of negative instances in the Web service taxonomy that were erroneously reported as positive, we can measure purity of a Web service taxonomy *within* subject categories. The false negative rate is the proportion of positive instances in the Web service taxonomy that were erroneously reported as negative, thus we can estimate purity of the Web service taxonomy *across* subject categories.

After obtaining performance measures, we used the t-test to examine whether the performance measures of different Web service taxonomies were significantly different. We conducted two t-tests with the performance measures according tothe sources of the input data set. Within each t-test, we compared two sets of performance measures : (1) the performance measures based on the WebServSOM generated Web service taxonomy (SOM^{θ}), and (2) the performance measures based on original taxonomy information from the corresponding Web service repository (BO or WO).

In our experiments, we intend to explore the performance of our proposed Web service taxon-

<Table 4> Summary of Hypotheses

Code	Hypothesis
H1 : Accuracy	
H1.1	The accuracy rates of two Web service taxonomies created withthe sample data set from *BindingPoint.com* are in the following order : $SOM^{\otimes} > BO$
H1.2	The accuracy rates of two Web service taxonomies created with the sample data set from *WebServiceX.net* are in the following order : $SOM^{\otimes} > WO$
H2 : False Positive Rate	
H2.1	The false positive rates of two Web service taxonomies created with the sample data set from *BindingPoint.com* are in the following order : $BO > SOM^{\otimes}$
H2.2	The false positive rates of two Web service taxonomies created with the sample data set from *WebServiceX.net* are in the following order : $WO > SOM^{\otimes}$
H3 : False Negative Rate	
H3.1	The false negative rates of twoWeb service taxonomies created withthe sample data set from *BindingPoint.com* are in the following order : $BO > SOM^{\otimes}$
H3.2	The false negative rates of two Web service taxonomies created with the sample data set from *WebServiceX.net* are in the following order : $SOM^{\otimes} > WO$

omy generation framework when compared to the existing Web service taxonomies collected from the two Web services repositories. The hypotheses we expect to prove are listed in <Table 4>.

4.2 Statistical Analysis and Discussion

Now we describe and analyze the results of our experiments. <Table 5> shows the effectiveness of different Web service taxonomies for each sample data set in terms of average accuracy, average false positive rate, and average false negative rate. We conducted paired-sample t-tests to analyze if there is any significant difference between the WebServSOM generated service taxonomies and the original Web service taxonomies from the Web services repositories.

The comparison between the original Web service taxonomies from *BindingPoint.com* (BO) and the WebServSOM generated service taxonomies ($SOM^{\otimes}$) in terms of accuracy, false positive rate, and false negative rate is summarized in <Table 6>. The results confirm that $SOM^{\otimes}$ achieved higher accuracy than BO, and the superiority of $SOM^{\otimes}$ is very distinct (.000 at the 0.05 level). It also shows that $SOM^{\otimes}$ has lower error rates (i.e., false positive rate and false negative rate) than BO. In other words, the WebServSOM generated service taxonomies achieved higher level of purity both within subject categories and across subject categories than did the original

<Table 5> Summary of Descriptive Statistics

Services Repository	Web Service Taxonomy	Accuracy	False Positive	False Negative
BindingPoint.com	WebServSOM ($SOM^{\otimes}$)	0.9860	0.010	0.004
	Original Category information (BO)	0.9577	0.036	0.006
WebServiceX.net	WebServSOM ($SOM^{\otimes}$)	0.9797	0.009	0.001
	Original Category information (WO)	0.8858	0.1135	0.000

<Table 6> Paired-Sample T-Test Comparison between *BindingPoint.com* and WebServSOM

	Mean	Std. Deviation	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
			Lower	Upper			
Acc	-0.028	0.016	-0.031	-0.025	-18.042	99	0.000
fp	0.026	0.015	0.023	0.029	17.774	99	0.000
fn	0.002	0.003	0.002	0.003	6.866	99	0.000

<Table 7> Paired-Sample T-Test Comparison between *WebServiceX.net* and WebServSOM

	Mean	Std. Deviation	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
			Lower	Upper			
Acc	-0.094	0.033	-0.101	-0.087	-28.117	99	0.000
fp	0.105	0.030	0.099	0.111	34.711	99	0.000
fn	-0.012	0.013	-0.014	-0.009	-8.697	99	0.000

Web service taxonomies from *BindingPoint.com*.

<Table 7> summarizes the comparison between the original Web service taxonomies from *WebServiceX.net* (WO) and the WebServSOM generated service taxonomies ($SOM^{\otimes}$) in terms of accuracy, false positive rate, and false negative rate. The results show that $SOM^{\otimes}$ achieved higher accuracy than WO, and the superiority of $SOM^{\otimes}$ is very significant (.000 at the 0.05 level). It also shows that $SOM^{\otimes}$ achieved a lower false positive error rate than WO, while WO outperformed $SOM^{\otimes}$ in terms of the false negative rate. This is not surprising because WebServiceX.net categorized Web services into only 7 general subject categories, which were not further specialized.

4.3 Summary of Findings

<Table 8> summarizes the results of testing the various hypotheses. The results of testing H1.1 and H1.2 show that the WebServSOM generated service taxonomies were significantly more accurate than the original Web service taxonomies collected from both *BindingPoint.com*

<Table 8> Results of Hypotheses Testing

H1 : Accuracy		p value	Result
H1.1	$SOM^{\otimes} > BO$	0.000 (α < 0.05, 2-tailed)	Confirmed
H1.2	$SOM^{\otimes} > WO$	0.000 (α < 0.05, 2-tailed)	Confirmed
H2 : False Positive Rate		p value	Result
H2.1	$BO > SOM^{\otimes}$	0.000 (α < 0.05, 2-tailed)	Confirmed
H2.2	$WO > SOM^{\otimes}$	0.000 (α < 0.05, 2-tailed)	Confirmed
H3 : False Negative Rate		p value	Result
H3.1	$BO > SOM^{\otimes}$	0.000 (α < 0.05, 2-tailed)	Confirmed
H3.2	$SOM^{\otimes} > WO$	0.000 (α < 0.05, 2-tailed)	Confirmed

and *WebServiceX.net*.

The results from testing H2.1 and H2.2 confirm that the WebServSOM generated service taxonomies achieved lower false positive rates than the original Web service taxonomies collected from both *BindingPoint.com* and *WebServiceX.net*. Since the false positive rate represents the purity level of taxonomies *within* subject categories, we conclude that each subject category in the WebServSOM generated service taxonomies contained less semantically irrelevant Web services *within* the subject categories than did those from both *BindingPoint.com* and *WebServiceX.net*.

The result of testing H3.1 shows that the WebServSOM generated service taxonomies achieved lower false negative rates than the original Web service taxonomies collected from *BindingPoint.com*. Based on the result, we concludethat more of the Web services in the WebServSOM generated taxonomies were correctly categorized to semantically corresponding subject categories than were those in *BindingPoint.com*.

The result from testing H3.2 confirms that the original Web service taxonomies achieved lower false negative rates than the WebServSOM generated service taxonomies did. We believe this could be attributed to the number of subject categories used in *WebServiceX.net*. This is not surprisingbecause the relatively small number of subject categories has a twofold effect : on the one hand, it would result in lower false negative rates; on the other hand, it would result in higher false positive rates. Consequently, we conclude that our Web service taxonomy generation framework generates more accurate Web service taxonomies than both existing Web services repositories.

5. Conclusion

In this paper, we have presented a novel clustering based Web service taxonomy generation framework, which can be used in conjunction with the existing Web service technology, such as WSDL, to support a more service discovery process. The feasibility and features of our proposed methodology have been demonstrated in a prototype system implementation. We have also reported on some preliminary evaluation results. This framework constitutes two components : iterative clustering analysis method and descriptive label generation method. The iterative cluster analysis method utilizes an unsupervised artificial neural network-based clustering algorithm as a core model to generate Web service taxonomy, while the descriptive label generation method semi-automatically assigns descriptive labels to subject category context in the Web service taxonomy by utilizing term frequency analysis.

The Web service taxonomies generated by this framework were empirically evaluated by comparing them with the existing Web service taxonomies in two different Web services repositories. The empirical evaluation showed that the outputs from our framework were significantly more accurate than the two Web service taxonomies from the two existing Web services repositories. We believe that this isone of the first attempts at applying unsupervised artificial neural network-based cluster analysis in the Web service domain. An extensive literature review reveals that, although various approaches have been proposed to apply data mining techniques in the Web service domain, no attempts have been made to manage domain-independent Web service taxonomies.

References

Dong, X., J. Madhava, and A. Halevy, "Similarity Search for Web Services", in *Proceedings of VLDB Conference*, Toronto, Canada, 2004.

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.

Hatzivassiloglou, V. and K. R. McKeown, "Towards the Automatic Identification of Adjectival Scale : Clustering Adjectives According to Meaning", in Proceedings of the 31st ACL. Columbus, Ohio, USA, 1993.

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., "Evaluation of Web Service Similarity Assessment Methods", *Journal of Intelligence and Information Systems*, Vol.15, No. 4(2009), 1~21.

Kohonen, T., *Self Organizing Maps,* Third ed. Berlin : Springer, 2001.

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 Retrieval*, New Orleans, Louisiana, United States, 2001.

Lewis, D. D., "Evaluating Text Categorization", in Proceedings of the Speech and Natural Language Workshop. San Mateo, CA, 1991.

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)*, 1999, 2034.

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

Sabou, M. and J. Pan, "Towards Improving Web Service Repositories through Semantic Web Techniques", *Web Semantics : Science, Services and Agents on the World Wide Web*, Vol.5, No.4(2007), 142~152.

Stroulia, E. and Y. Wang, "Structural and Semantic Matching for Assessing Web-Service Similarity", *International Journal of Cooperative Information Systems*, Vol.14, No.4(2005), 407~437.

Swets, J. A., Effectiveness of Information Retrieval Methods, American Documents, 1969.

Yang, Y., "An Evaluation of Statistical Approaches to Text Categorization", *Information Retrieval*, 1), 1999, 69~90.

Zhuge, H. and J. Liu, "Flexible Retrieval of Web Services", *The Journal of Systems and Software*, Vol.70(2004), 107~116.

Abstract

인공신경망 기반 웹서비스 분류체계 생성 프레임워크의 실증적 평가

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 최근의 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. 웹서비스를 서비스 중심 컴퓨팅환경으로서 운용하기 위해서는 웹서비스 저장소가 완성도 높게 조직화되어 있어야 할 뿐 아니라, 사용자들의 필요에 맞는 웹서비스 컴포넌트를 찾을 수 있는 효율적인 도구들을 제공하여야 한다. 서비스 중심 컴퓨팅을 위한 웹서비스의 중요성이 증대됨에 따라 웹서비스의 분류체계를 효율적으로 제공할수 있는 기법의 수요 또한 증대된다. 다수의 웹서비스 저장소들은 웹서비스 분류체계를 제안하여 왔지만, 대부분의 분류체계는 활용하기에는 제대로 발달하지 못하였거나 관리하기에 너무 어려운 단점을 갖고 있다.

이 논문에서는 인공신경망 기반 군집화 기법과 XML 기반의 웹서비스 기술표준인 WSDL의 의미적가치를 활용하여 웹서비스 분류체계 생성 프레임워크를 제안한다. 이 논문에서 인공신경망을 활용하여 제안하는 웹서비스 분류체계 생성 프레임워크를 프로토타입 시스템로 개발하였으며, 실제 운용되고 있는 웹서비스 저장소로부터 획득한 실제 웹서비스들을 사용하여 제안하는 웹서비스 분류체계 생성 프레임워크를 실증적으로 평가하였다. 또한 제안하는 방식의 효용성을 보여주는 실험결과를 보고한다.

Keywords : 웹서비스, 분류체계, 데이터 마이닝, 인공신경망, 프로토타입 시스템, 군집화, 평가

* 서울시립대학교 경영대학 교수

저 자 소 개

황유섭

현재 서울시립대학교 경영대학 교수로 재직 중이다. The University of Arizona에서 경영정보시스템을 전공하여 경영학사, 석사, 그리고 경영학 박사학위를 취득하였다. 미국 NASA와 Raytheon의 Hydrology Resource Management Project에 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.