
© Copyright 2008 Idea Group Inc. / IGI Global
This chapter appears in "Research and trends in data mining technologies and applications" > edited by David Taniar > Copyright 2008, IGI Global, www.igi-pub.com. Posted by permission of the publisher.
Chapter XII

Facilitating and Improving the Use of Web Services with Data Mining

Richi Nayak, Queensland University, Australia

Abstract

Web services have recently received much attention in businesses. However, a number of challenges such as lack of experience in estimating the costs, lack of service innovation and monitoring, and lack of methods for locating appropriate services are to be resolved. One possible approach is by learning from the experiences in Web services and from other similar situations. Such a task requires the use of data mining to represent generalizations on common situations. This chapter examines how some of the issues of Web services can be addressed through data mining.
Introduction

Web services have received much attention in recent years (Clark, 2002; Kearney et al., 2002; Booth et al., 2004). By allowing applications residing in disparate systems to communicate, Web services create myriad opportunities for implementing new business models based on the offering and consumption of services. However, despite its apparent benefits, businesses are slow in the widespread adoption of Web services (Chen, 2003). This is largely due to the many issues relating to the technology, as well as the lack of experience in this area. These issues span across the entire development process, from initial planning through to final implementation. Additionally, as the number of Web services increases, it becomes increasingly important to provide a scalable infrastructure of registries that allows both developers and end users to perform service discovery. One possible approach to overcome these problems is by learning from the experiences in Web services and from other similar situations. Such a task leans itself towards the use of models that represent generalizations on common situations. The high volume of data accumulated from Web services activities is an excellent source for building these models. What is needed therefore is an automated mechanism to perform this task. This is where data mining (DM) plays an important role.

But what are the sorts of knowledge that can be extracted to alleviate the above problems? What data is available for data mining? How can we extract the useful information from this data? Since this is an innovative application area, what are some of the issues and challenges that we may have to face and resolve? These are the questions this chapter attempts to answer. In this chapter, first we explain the Web services enabling technologies and issues affecting data mining. Next, we recommend a set of applications that can leverage problems concerned with the planning, development, and maintenance of Web services. We also summarize some of the existing works that have been conducted for the mutual benefits of Web services and data mining technologies. We end the chapter by discussing the future of these two widely adopted technologies.

Web Services Environment

Web services emerged from the fields of distributed computing and the Web. Web services operate with existing Internet protocols to provide Web-based access, easy integration, and service reusability by using loosely coupled connections, and vendor, platform, and language independent protocols (Systinet Corp, 2002; Booth et al., 2004). Web services enable applications residing in disparate systems...
to communicate and produce interesting interactions. Based on the existing Web protocols and eXtensible Markup Language (XML) standards (Yergeau, Bray, Paoli, Sperberg-McQueen, & Maler, 2004), the Web services architecture makes it possible for businesses to use small pieces of software as services to compose complex operations that facilitate its business processes, using the Internet as the underlying infrastructure.

**Enabling Technologies**

Web services are implemented by a set of core technologies that provide the mechanisms for communication, description, and discovery of services. The standards that provide these functionalities are simple object access protocol (SOAP), Web Services Description Language (WSDL), and universal description, discovery, and integration (UDDI) (Vasudevan, 2001; Curbera et al., 2002). These XML-based standards use common Internet protocols for the exchange of service requests and responses. Figure 1 shows the relationship of these technologies as a standards stack for Web services.

When a service provider creates a new service, it describes the service using WSDL. WSDL defines a service in terms of the messages to be exchanged between services and how they can be bound by specifying the location of the service with a URL. To make the service available to service consumers, the service provider registers the service in a UDDI registry by supplying the details of the service provider, the category of the service, and technical details on how to bind to the service. The UDDI registry will then maintain pointers to the WSDL description and to the service. When a service consumer wants to use a service, it queries the UDDI registry to find a service that matches its needs and obtains the WSDL description of the service, as well as the access point of the service. The service consumer uses the WSDL description to construct a SOAP message to be transported over HTTP with which to communicate with the service.

**Figure 1. Web services standards stack**

| Service publication and discovery (UDDI) | Core Web services standards |
| Service description (WSDL) |
| Service communication (SOAP) |
| Packaging (XML) |
| Transfer and network protocols (HTTP & TCP) |
Since HTTP is used, traffic for Web services is directed to Web servers. As a result, all communications between Web services pass through the Web server. Since Web servers store access request information in Web server access logs, all interactions between Web services will thus be recorded, similar to the human-server interactions that are prevalent on the Web now. Web server access logs recording site access have been a rich source for discovering the browsing behavior of site visitors (Wang, Huang, Wu, & Zhang, 1999; Srivastava, Cooley, Deshpande, & Tan, 2000; Nayak, 2002). In a similar manner, logs that record Web service access can be mined for Web service usage patterns by organizations. With the huge volume of access, the amounts of logs that can be collected make a feasible source for mining operations.

**Issues in Web Services**

**Cost Planning**

Chen (2003) performed an analysis on factors that are affecting the adoption of Web services. Among the identified decision criteria that hold constant across all cases are financial considerations. Costs that can be saved both in the short and long term are one of the major reasons that businesses can justify their investments in a new technology. Therefore, unless businesses have an idea of how much they should spend on Web services, and on the foreseeable savings, they will be indecisive on the adoption of Web services. The enabling technologies of Web services are only concerned with the technicalities, and do not have any provision for the planning of costs and savings that can gained.

Answering to this is the cost and performance model for Web services investment (Larsen & Bloniarz, 2000). The model helps businesses assess the costs in terms of the functionalities required and increases in target performance. For each assessment area, three levels of service (modest, moderate, and elaborate) are defined in order to identify the most profitable option. Worksheets setting out the details of each of these areas are used as aids for estimating the figures.

Although the model provides tools for estimating costs, it does not take into consideration that many businesses may not have any experience in Web services and thus find the estimation difficult in the first place. Furthermore, it does not provide any indications of whether the benefits from the proposed project match those of competitors’ projects that are similar. That is, there is no measure on the effectiveness at which a business can deploy Web services. If a business can find out that other similar projects cost less but yield the same returns, then it may consider outsourcing the project instead of developing it in-house.

A distinctive characteristic of DM is that the outputs are generalizations of the inputs (Fayyad, Piatesky-Shapiro, Smyth, & Uthurasamy, 1996). This makes it an
ideal tool for benchmarking, especially in predictive modeling operations. Using the derived models and rules, an analysis can feed in experimental data and get results that represent what the majority would get if the data were real. Based on this, if models representing Web services deployment costs can be built, then businesses intending to adopt Web services can make use of these models both for estimating the cost of their deployment, as well as deciding whether outsourcing them is more feasible than developing them in-house.

**Service Discovery**

The discovery of services requires seven steps (Hoschek, 2002): description, presentation, publication, request, discovery, brokering, and execution. Description is the process of defining metadata for a service, such as the interfaces it offers, the operations and arguments for an interface, and the binding of network protocols and endpoints. Presentation is concerned with mechanisms that allow the retrieval of the service descriptions. This requires a means of uniquely identifying a service in a global scale. Publication makes a service known to interested clients via the use of a registration system. Request and discovery involves a user formulating a request for a service, while discovery involves searching the registry for candidate services that implement the requested functions. Brokering and execution are concerned with the scheduling and allocation of resources for the execution of the services.

WSDL and the UDDI are two standards designed for the description, presentation, publication, request, and discovery of Web services. However, they have limitations in addressing the problem. The UDDI especially has drawn much criticism. Although WSDL facilitates the description of the service parameters, messages to be exchanged between applications, and how to connect to the services, it provides little clue to the service consumer as to what the service does. The only parts of the description that may give hints on the functionalities of a service are the name attributes of the parameters, messages, and operations. This lack of semantics led to the development of DARPA Agent Markup Language (DAML)-based languages (http://www.daml.org/services/) for service description, where service capability matching is based on the inputs, outputs, preconditions, and effects, and ontology is used to encode the relationship between concepts (McIlraith, Son, & Zeng, 2001; Paolucci, Kawamura, Payne, & Sycara, 2002). However, with the current state of Web services, we are still a long way from automatic service matching. For now, the manual discovery of services will have to suffice, and effort is needed to improve its efficiency.

Similar to WSDL descriptions, the UDDI does not make use of semantic information to describe and discover services based on their capabilities (Paolucci et al., 2002). Apart from the text descriptions, there is no provision for specifying the capabilities of the service. The categorization of businesses and services—that is, the “yellow
pages”—is also of little help in the discovery of suitable services (Newcomer, 2002). The classification schemes used—NAICS and UNSPSC—were designed for the broad-based classification of industries, products, and services. There is little or no differentiation between products or services in the same line of business. These schemes do not provide the specificity needed for service discovery. Furthermore, searching in UDDI is restricted to keyword matching on names, locations, business, bindings, and tModels (unique identifiers for reusable concepts). There is no provision for inference or flexible match on the keywords, which means service providers and requesters must choose the names and description of services very precisely when using the UDDI in order to be found.

Normally, developers have to search the service from tens or hundreds of service entries in UDDI. This process is very time consuming and can be considered as loss of profits in terms of business. As the single most pervasive source for the discovery of Web services, there is a great need to address the shortcomings of the UDDI. A most intuitive way to help alleviate the situation is by predicting what a service requester may be looking for and suggesting services accordingly. This can be done by extracting common behaviors of users searching the UDDI by mining the user query logs. Thus data mining can help Web search engines to find high-quality Web pages and enhances Web click stream analysis.

Application of Data Mining in Web Services

The benefits of the data mining applications can be seen as those that deliver business value and those that have technical value (Nayak & Tong, 2002). Data mining applications can be used by management to assist in making strategic decisions or by human resource personnel in maximizing staffing levels while minimizing costs. More specifically, data mining can be used to provide insights on the planning of Web services deployment via “Web services cost and savings prediction,” and how costs on staffing for the monitoring of services can be optimized via “performance monitoring.” Data mining applications can also be used by technical staff in devising new services or in services that their organization can use via “services innovations and recommendations.”

Web Services Cost and Savings Prediction

It is difficult for businesses to gauge the costs and savings of a Web service deployment with having little or even no experience in deployments. However, businesses can learn from the experiences of similar organizations and get a good approxima-
tion of these values from the data collected by research firms such as Nemertes (Johnson, 2003).

Value prediction is suitable in this instance to model the investment vs. return functions for the prediction of figures for costs and savings. Regression techniques derive the predicted continuous values obtained from functions that best fit the case (Devore, 1995). For predicting the costs, the input data required consists of, for each deployment, the number of staff members involved, the time it took, and the complexity of the deployment. The complexity of the deployment can be quantified in terms of the lines of code used in the programs, and the annual revenue from the operations that the deployment oversees. The costs of the proposed deployment can be predicted based on these parameters.

Once the costs are known, prospective savings can be predicted. Using inputs such as the cost of the deployment, and the original and new cost of the operation, savings can be determined. Having determined the costs and the savings that can be gained, the return of investment for Web services deployments can be calculated based on these figures. Businesses can then identify the size of Web services deployment that is best suited for them and turn the discovered insights into action.

**Performance Monitoring**

Strategic placement of human resources plays a crucial role in the effective monitoring of performance and handling of events. This leads to the need to prioritize tasks. A service being used by many clients at the time when a problem occurs should have a higher priority than a service being used by few clients at the same time. By knowing the usage pattern of services, training programs on groups of services with similar usage patterns can be developed. This allows staff monitoring the services at certain times to have a more in-depth knowledge of particular services.

To identify services with similar usage patterns, similar time sequence analysis can be used (Peng, Wang, Zhang, & Patker, 2000). The input for such an operation is time-series data recording the number of clients using a particular service at any moment in time. Although such data is not normally collected explicitly, it is implicitly recorded in the Web server access logs. The steps in generating this time-series data from Web server logs are as follows:

1. Select from the Web server log all entries related to the offered Web services by extracting all entries containing the Web service’s URL in the URL field.
2. Group the entries by Web services and client IP addresses, and then order the entries by time. This gives a set of a client’s interaction with a Web service.
3. Calculate the time between each interaction to determine separate client sessions with the Web service. A client session is one ‘use’ of the Web service. The duration of a client session for different services varies depending on the nature of the service. Setting the threshold of session boundaries thus requires the knowledge about the individual services.

4. For each service, count the number of clients using it at specified time intervals. This can then be used to construct the time-series graph for each service.

Algorithms for approximate subsequence matching in time-series (Han & Kamber, 2001) now can be applied to find Web services that have similar usage patterns. These patterns can then be used to help in the design of roster schedules that optimize staffing levels and skill requirements while minimizing the number of employees that need to be present.

**Service Innovation**

It is important for service providers to establish themselves in the market by offering a range of quality services. The set of queries used by potential clients to find suitable Web services is a rich source for finding clues about what the clients want. If an unusual search term is used with other common search terms in the queries, and if the search terms are all related, then it is a good indication that there is a demand for a new service. The unusual search term may represent a new concept or a specialization of a general service currently being offered. As an example, short message service (SMS) sends text messages to mobile phones, while a more recent technology, multimedia message service (MMS), sends multimedia messages. SMS is a frequently used search term, but MMS is not. As the technology becomes more prevalent, demand for MMS Web services will emerge, and the appearance of MMS in query data will be evidence of this.

The simplest approach in discovering uncommon search terms is by deviation analysis (Devore, 1995). Having counted the frequencies of the search terms appearing, simple measures such as median, quartiles, and inter-quartile range (IQR) can be calculated. Then using the common heuristic that outliers fall at least 1.5’ IQR above the third quartile or below the first quartile, the unusual search terms can be identified. An alternative measure is the use of support to count the number of times the term appeared in total terms. If a search term has very low support, then it can be classified as an outlier.

Given that the demand for different services varies, applying these measures to the raw frequency count will produce biased results towards less popular services, producing many false positives. This is best illustrated using an example. A popular service is searched 1,000 times using a common search term $Q_1$ and 10 times using
an uncommon search term \( Q_2 \). A very specific service aimed at a niche market is searched seven times using \( Q_3 \) and three times using \( Q_4 \), both of which are common for the service. When search terms for all the services are taken into account, and statistics is applied to the data, \( Q_2 \), \( Q_3 \), and \( Q_4 \) will be identified as uncommon search terms. However, \( Q_3 \) and \( Q_4 \) are false positives because they represent 70% and 30% of searches for the service. On the other hand, although \( Q_2 \) has 10 occurrences, it is only 1% of all searches for the popular service. Clearly, \( Q_2 \) is the only outlier that should be identified in this case. Extrapolating this to a real scenario, one can expect to find no true outliers for the popular services, which is far more important than outliers for less popular services. The solution to this is to group searches into search areas and then find the outliers for each search area. This can be done by:

1. Grouping the queries into search sessions
2. Joining all search sessions that are similar to form search areas
3. Form search term pools for each search area
4. Within each search term pool, apply statistics to find the uncommon search terms that suggest demands for a new Web service

**Service Recommendation**

The location of Web services using existing Web services search engines (keyword based) can be a lengthy endeavor. This method of service discovery suffers from low recall, where results containing synonym concepts at a higher or lower level of abstraction to describe the same service are not returned. Web service providers can recommend services to clients based on the services that other similar clients have used in the past with the use of DM, or by returning an expanded set of results to the user with the use of ontology (Beeferman & Berger, 2000; Bernstein & Klein, 2002; Wen, Nie, & Zhang; Yu, Liu, & Le, 2004). This is because similar clients are likely to have similar service needs. A single data mining operation or in combinations can be used to achieve this.

**Based on Predictive Mining**

Service providers have information such as the line of business, size of business, and what services their clients use. These can be used as inputs for predictive modeling operations and recommendations can be made to new clients. Inputs such as the interfaces, functionality, and security offered by the service, as well as the cost, and other resources required by the service can also be considered in analysis. Classification techniques such as decision trees can be used to build rules.
on service subscriptions. Since the only information service providers have about clients are those for billing purposes, the number of attributes available is small. Consequently, the structure of the resulting decision tree will be relatively simple and easily comprehensible to a human analyst. To further enhance the success rate of recommendations, service providers can find dissociations among the services they offer. Dissociations capture negative relationships between services with rules such as $X \Rightarrow Z; X \land Y \not\Rightarrow Z$—that is, the use of services $X$ and $Y$ implies that it is unlikely service $Z$ will also be used, even though $X$ and $Z$ are often used (Teng, 2002). By incorporating these dissociations in the recommendation process, more specific recommendations can be made.

**Based on Association Mining**

Web server access logs record all interactions between Web services and users. With the huge volume of Web service access, the amount of logs that can be collected makes a feasible source for identifying Web services with similar usage patterns. The tasks are: (1) selection from the Web server log all entries related to the offered Web services, (2) extraction of a set of a client’s interaction with a Web service, (3) calculation of client sessions with the Web service, and (4) application of an association mining algorithm to find Web services with similar usage patterns.

**Based on Clustering Mining**

While previous approaches capture the intra-query relationships by clustering queries on a per query basis (Beeferman & Berger, 2000; Wen et al 2002), they omit the inter-query relationships that exist between queries submitted by a user in one search session. A better option is to group the similar search sessions and provide suggestions of search terms from search sessions that belong to the same cluster.

The first task is to consolidate the data from the user query and Web server logs. This is done by matching the query recorded in the query log with the subsequent service descriptions viewed by the user recorded in the Web server log. The next task is to form search sessions to arrange a set of queries in sequence by a user to locate a particular service. Search session similarity now can be calculated based on the similarity of the set of search terms used and the set of service descriptions viewed between two search sessions. The Jaccard coefficient (Han & Kamber, 2001) can be used to calculate the similarity of the search terms and service descriptions sets. Search sessions are assigned to the same cluster if they have many queries and service descriptions that are the same from the entire query and service description pool. The type of clusters desired is therefore globular in nature. Also the algorithm must be resistant to noise and outliers. The agglomerative hierarchical clustering
(Karypis, Han, & Kumar, 1999) is well suited to generate these types of clusters. After the clusters are formed, the support for each of the search terms in each cluster is counted and then assigned weights. The weights can be used to predict a user’s service need by suggesting search terms from the cluster with the largest weight for the user’s search term. Depending on the size and number of search terms that make up the clusters, the suggested terms can either be all search terms within the cluster, or be limited to those from a predefined number of most similar search sessions. A test was conducted for evaluating the effect of the measure combining both keyword and service description similarity. The results show that the search sessions clustered using the combined measure is more similar internally and thus the clusters are more compact. This is essential in the suggestion of search terms, as users would only be interested in suggestions that are highly similar to those submitted.

**Existing Work**

This section discusses the existing work that crosses paths between data mining and Web services.

**Use of Data Mining in Improving Web Services Usage**

The previous section discusses a number of possible applications of data mining to assist the Web services. In this section we outline some existing works. The majority of work is in the direction of addressing the shortcoming of UDDI by finding relationships between search terms and service descriptions in UDDI.

Sajjanhar, Jingyu, and Yanchun (2004) have applied the regression function called singular value decomposition (SVD) to discover semantic relationships on services for matching best services. Their preliminary results show a significant increase in correct matching between service descriptions and the search terms after application of their algorithm with IBM UDDI. The matched results are not merely based on the number of matched keywords within the service descriptions. The algorithm evaluates the keyword global weights within the SVD procedure and aggregates services containing the highest global weight words to find semantic matched services. Wang and Stroulia (2003) developed a method for assigning a value of similarity to WSDL documents. They use vector-space and WordNet to analyze the semantic of the identifiers of the WSDL documents in order to compare the structures of their operations, messages, and types, and to determine the similarity among two WSDL documents. This helps to support an automatic process to localize Web services by distinguishing among the services that can potentially be used and that are irrelevant to a given situation. Dong, Halevy, Madhavan, Nemes, and Zhang (2004) build a
Web service search engine to support the similarity search for Web services along with keyword searching with utilizing clustering and association mining. Starting with a keyword search, a user can drill down to a particular Web service operation. However, when unsatisfied, instead of modifying the keywords, the user can query for Web service operations according to the most similar and semantically associated keywords suggested by the engine using the data mining techniques.

Gombots et al. (2005) attempt to apply data mining to Web services and their interactions in order to analyze interactions between Web service consumers and providers. They toss a new term, “WSIM—Web services interaction mining,” to analyze the log data to acquire additional knowledge about a system. They identify three levels of abstraction with respect to WSIM: the operation level, the interaction level, and the workflow level. On the Web service operation level, only one single Web service and its internal behavior is examined by analyzing a given log output of the Web service. On the Web services interaction level, one Web service and its “direct neighbors” Web services (that the examined WS interacts with) are examined. This analysis reveals interesting facts about a Web service’s interaction partners, such as critical dependencies. On the highest level of abstraction—the Web service workflow level—the large-scale interactions and collaborations of Web services which together form an entire workflow are examined. This details the execution of the entire process: what is the general sequence of execution of various operations?

Malek et al. (2004) apply data mining in security intrusions detection while the Web services are in use. They show the impact of mining in detecting security attacks that could cripple Web services or compromise confidential information. They determine the relevance of different log records and define the attack signature with the use of sequential pattern mining with logs. Then they discover the highly compact decision rules from the intrusion patterns for pattern searching that help to describe some safeguard against the attacks.

Use of Web Services Technologies to Develop Data Mining Solutions

Web services techniques are increasingly gathering data from various sources that can be used for mining. Additionally, data mining tools are now required to access a variety of standards and platforms. The solutions for interoperability by using XML and SOAP as means of Web services communication can assist data mining by standardizing importing data and information to XML format. Web services can offer assistance to data mining in integration of data coming from various sources. The SOAP protocol enables data interaction on the Web and therefore makes the
collection of data possible (Nayak & Seow, 2004). Some efforts have been made to implement these protocols, but in fact the full potential of these technologies has not yet been reached. An example is Web services for DB2 Intelligent Miner (http://www.alphaworks.ibm.com/tech/ws4im), “a collection of Web services that allow clients to describe and perform basic mining tasks using XML, XML Schema, and XPath on top of DB2 Intelligent Miner.”

Another example of a SOAP-based data mining solution is XML for Analysis (http://www.xmla.org/tdwievent.asp), an open industry-standard Web service interface for online analytical processing and data mining functions. It provides “a set of XML Message Interfaces that use SOAP to define the data access interactions between a client application and an analytical data provider.” The interfaces are aimed at keeping the client programming independent from the mechanics of data transport, but at the same time providing adequate information concerning the data and ensuring that it is properly handled. This data mining solution is platform, programming language, and data source independent.

There are many data mining applications that use Web services technologies to implement them efficiently. An example is online banking, which has recently grown substantially as a Web service. As the online transactions increase, so does the possibility of fraud, in particular credit card fraud. Chiu and Tsai (2004) proposed a Web services-based collaborative scheme for participating banks to share their knowledge about fraud patterns. The participating banks share their individual fraud transactions and new transactions via a central location. The data exchange in this heterogeneous and distributed environment is secured with WSDL, XML, and SOAP. The frequent pattern mining is then applied on this integrated data for extracting more valuable fraud patterns to improve the fraud detection.

Other research introduces a dynamic data mining process system based on Web services to provide a dynamical and satisfied analysis result to the enterprise (Chiu & Tsai, 2005). Each data mining process (data pre-processing, data mining algorithms, and visualization analysis) is viewed as a Web service operated on the Internet. The Web service for each activity provides its functionality. Depending on the user’s requirement, the Web services are dynamically linked using the Business Process Execution Language for Web Service (BPEL4WS) to construct a desired data mining process. Finally, the result model described by the Predictive Model Markup Language (PMML) is returned for further analysis. PMML is an XML markup language defined for data mining functions and models (Wettschereck, & Muller, 2001) to make easy and efficient data models and result interpretation.

There are many application-oriented research studies as well. Zheng and Bouguettaya (2005) model the biological entities and the dynamic processes as Web services, and then propose a Web service mining approach to automatically discover the unexpected and potentially interesting pathways.
Future Directions: Semantic Web and Ontologies

The Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) is described as the next generation of Web architecture. It operates on new markup language standards such as Web Ontology Language for Services (OWL-S) (http://www.w3.org/Submission/OWL-S) and DARPA Agent Markup Language Semantic Markup extension for Web Services (DAML-S). OWL-S and DAML-S are high-level ontologies at the application level meant to answer the what and why questions about a Web service (Alesso & Smith 2004). An ontology defines the relations among terms (Maedche, 2003). The use of ontologies can relate the information on numbers of Web services to the associated knowledge structures and inference rules in a Semantic Web.

The Semantic Web with ontology is not merely an ordinary repository or normal online meta-data, but it becomes the intelligent Web having automated reasoning. An ontology-based Semantic Web describes its properties and capabilities so that: (1) software can automatically determine its purpose, thus automating service discovery; and (2) software can verify and monitor service properties, thus automating service monitoring. As the future of Web services greatly depends on their ability to automatically identify the Web resources and execute them for achieving the intended goals of the user as much as possible, OWL-S and DAML-S can achieve this, whereas UDDI cannot.

Recently, some preliminary works have been conducted to employ Web semantic techniques, showing its applicability in the concrete description of the Web resources. Bernardi et al. (2004) propose the process models to be described as first-order ontologies, and then habilitate the automatization of the searching and composition of Web services. Mandell and MacIlraith (2004) present an integrated technology for the customized and dynamic localization of Web services using the Business Process Language for Web Service (BPWS4J) with the semantic discovery service to provide semantic translation to match the user requirements. Soyaden and Singh (2004) developed a repository of Web services that extends the UDDI current search model. The repository in the form of ontology of attributes (based on DAML) provides a wide variety of operations such as the publication of services, costs of services, and service selection based on functionality. Similarly, Li, Yang, and Wu (2005) proposed an approach of ontology use in e-commerce service search based on generated query and decision-making process. The high-level ontology (based on DAML) positioning above WSDL relates service description of a WSDL document to descriptions of other WSDL documents. Benatallah et al. (2004) propose a matching algorithm that takes as input the requirements to be met by the Web services and an ontology of services based on logic descriptions, and recommends the services that best comply with the given requirements.
However, these approaches still require data mining to improve the user’s search satisfaction. The semantic component must provide a mechanism for effectively classifying and selecting Web services based on their functionality and supporting the search of WSDL descriptions (in the form of tModel) of selected Web services in a non-sequential order within the directories or registries. So the requirements may be expressed by the user in terms of the functionality of the Web service needed instead of, for example, the name of the service, the name of the organization that provides the service, or the categories specified in an UDDI.

Data mining from the Semantic Web emphasizes the usage of Semantic Web technologies for mining purposes such as the usage of taxonomies in recommender systems, applying association rules with generalizations, or clustering with background knowledge in form of ontology. The applications may use association rules to discover relations on service descriptions in WSDL documents; for example, users who invoke a shopping service may possibly invoke another service in the area of banking. Furthermore, the structure of ontology can be implemented by a data mining application using trees. By defining each node in the tree as a service description of a WSDL document, each tree path can form an ontology that describes the similarity of a path with other paths.

Conclusion

In this chapter, we have discussed the domain of Web services and applications of data mining techniques in facilitating and improving the use of Web services. The data mining tasks that find applications in Web services mining include value prediction, similar time sequence analysis, deviation analysis, classification, clustering, and association mining. These applications range from delivering business value that can be used by management for strategic decision making, to providing technical benefits that target specialist end users. Further testing is required to identify the real value of the applications. Additionally, because some applications such as search term suggestion require real-time responses, techniques for providing results efficiently need to be developed. These may include new algorithms for scheduling the processing of requests and delivery of responses to multiple users so the information is returned as at close to real time as possible.

With some existing works, we show that the data mining techniques play an important role as the emerging technologies in Web service discovery and matching. On the other hand, as data mining applications built on Web services become more popular, there is growing need for further research to develop Web services-based data mining solutions which can scale the large, distributed data sets that are becoming more popular. Since HTTP, XML, and SOAP are platform independent, it is hoped that
this will contribute to solving the blockage that has occurred among the competing proprietary protocols in the area of data mining. An example is a real-time Web services-based collaborative scheme involving data mining techniques to assist in detecting credit card fraud in online banking services.

The future holds for data mining techniques to apply with Semantic Web services and ontology to automate Web service discovery processes.

**Acknowledgment**

I would like to thank Cindy Tong and the ITB239, 2005 semester 1 students for assisting me in conducting the literature review on Web services and data mining usage.

**References**


