Web Access of Meteorological and Oceanographic Data

Frederick E.Petry[1], Bruce Lin[1], Ricahrd Sween[1] and Roy Ladner[2]
[1]Naval Research Laboratory, Mapping, Charting and Geodesy
Stennis Space Center, MS 39529 USA
[2] Naval Meteorology and Oceanography Command
Stennis Space Center, MS 39529 USA

Abstract-: **Web Services are becoming the standard technology used to share data for many Navy and other DoD operations. These enable an automated capability to obtain and integrate data for data fusion. However assimilation of data from Web-based sources means that differences in schema and terminology prevent simple querying and retrieval of data. Thus, machine understanding of the Web Services interface is necessary for automated selection and invocation of the correct service. In this paper we describe an advanced architecture that can provide access to web-based meteorological and oceanographic (METOC) data that can be utilized in geospatial data fusion. We also discuss the use of case-based classification as an alternative/ supplement to using ontologies for resolving knowledge sharing. While ontologies encompass a formal definition of a domain of interest, case-based reasoning is a problem solving methodology that retrieves and reuses decisions from stored cases to solve new problems, and case-based classification involves applying this methodology to classification task.**

I. Introduction

Information is now as important as tanks, ships and aircraft in today's military. Rapid access to data and the ability to share data are seen as significant to gaining superiority over opposing forces [1]. Web Services are becoming the technology used to share data for many Navy and other DoD operations. Web Services technologies provide access to discoverable, self-describing services that conform to common standards. Thus, this paradigm holds the promise of an automated capability to obtain and integrate data. However, the automated integration of applications to access and retrieve data from heterogeneous sources in a distributed system such as the Internet poses many difficulties. Assimilation of data from Web-based sources means that differences in schema and terminology prevent simple querying and retrieval of data. Machine understanding of the Web Services interface is necessary for automated identification, selection and invocation of the correct service. Service availability must also be resolved.

There has been considerable work on ontologies to help resolve these difficulties so as to share knowledge among various domains of interest. Ontologies describe a formal definition for a domain of interest through the terms and concepts of the domain and their interrelationships, and support automated computer reasoning on a domain through specification of its content. In some uses of ontologies, Web Services data providers are presupposed to deploy an ontological description of their Web Service to support automated discovery and integration by interested client applications [2]. Our approach does not require such descriptions.

There has been some research on Web Service classification as a means of automating or semi-automating the annotation of Web Services with semantic meaning. That work has had as its focus the automatic generation of Web Services ontologies such as OWL-S [3, 4].

In this paper we depart from the exclusive use of ontologies and examine the direct use of case-based classification as an alternate approach to support automated discovery of meteorological and oceanographic Web Services. Case-based reasoning (CBR) is a problem solving methodology that retrieves and reuses decisions from stored cases to solve new problems, and case-based classification focuses on applying CBR to supervised classification tasks. This approach generalizes well in sparse data, which characterizes our Web Services application. Unlike ontologies, case-based classification does not require formal domain definition and its use does not require data providers to deploy any additional specialized descriptions of their Web Service.

We are currently developing an Integrated METOC Broker (IWB) for the US Navy. Its objective is the automated discovery and application integration of meteorological and oceanographic (METOC) Web Services. We are examining the use of case-based classification in the IWB to support automated Web Services discovery.

The remainder of this paper is organized as follows. First, we briefly overview Web Services and previous work on ontologies in support of automated data exchange. Following this, we describe our work on the IWB. We then explain our approach for classifying METOC Web Services using a case-based classifier. We close with a discussion of future research goals.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE
Web Access of Meteorological and Oceanographic Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory,Mapping, Charting and Geodesy,Stennis Space Center,MS,39529

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002176. Presented at the MTS/IEEE Oceans 2008 Conference and Exhibition held in Quebec City, Canada on 15-18 September 2008. U.S. Government or Federal Rights License.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

II. Web Services and Ontologies

Web Services provide data and services to users and applications over the Internet through a consistent set of standards and protocols such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), the Web Services Definition Language (WSDL), and Universal Discovery Description and Integration (UDDI). XML has become one of the widely used standards in interoperable exchange of data on the Internet but does not define the semantics of the data it describes. XML Schemas define XML documents through structures that describe elements, attributes and data types, among others [5]. WSDL describes the acceptable requests that will be honored by a Web Service, the types of responses that will be generated [6], and the XML messaging mechanism of the service. For example, the messaging mechanism may be specified as SOAP. A UDDI registry provides a way for data providers to advertise their Web Services and for consumers to find data providers and desired services. An interface to a UDDI registry may allow users to search for Web Services by business category, business name, or service [7]. This advertisement of Web Services may not be desirable for net-centric operations in the DoD community.

Interacting with multiple Web Service interfaces poses issues for client application integration and maintenance. Addressing these issues may involve adoption of a single, uniform Web Service interface that may be implemented by multiple diverse data providers within a community. This may be found, for example, within the METOC community of interest where the Joint METOC Broker Language (JMBL) has been specified as the web service interface for METOC data exchange within the Department of Defense [21]. However, even where data providers have conformed to a recognized interface standard, custom coding to integrate applications with the interface remains necessary.

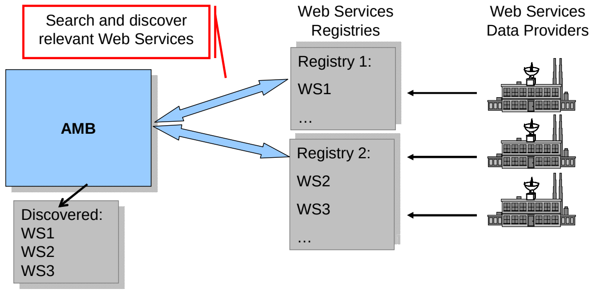
Recent efforts to improve interoperability include Web Services technologies such as WSDL and XML Schemas. While these provide structured content, their semantics are limited and not designed for interoperability (i.e., they may employ different meanings for the same terms or the same meanings using different terms, each of which limits their interoperability). Ontologies are often considered to be the basis of semantic meaning for these sorts of documents. Ontologies define the terms and concepts used to represent knowledge in a given domain of interest. They provide the structures that capture the relationships among concepts and enable applications to reason over them. Ontological frameworks for describing the semantics of data include such developments as the Resource Description Framework (RDF) and Web Ontology Language (OWL). RDF provides a flexible representation of information and a reliable means of supporting machine reasoning [8]. OWL permits users to more fully describe the meanings of terms found in Web documents and to represent the relationships among these terms [9].

Numerous methodologies for engineering and maintaining domain ontologies have been reported [10]. In some approaches, the starting point for ontology development is the specification of the questions the ontology should answer and/or problems it should solve. Generally, strategies for domain knowledge acquisition may vary from bottom-up to top-down. There are also editors that assist with ontology development, such as the open source editor Protégé. A Protégé extension supports OWL ontologies [11]. Even with these tools, ontology development remains a time- and skill-intensive activity.

OWL-S extends OWL to supply the constructs for defining an ontology of services that is intended to support automated Web Services discovery, invocation, and composition. For example, a Web Services provider could advertise its services in OWL-S in a service registry, where software agents or brokers could discover it through querying. The software agent or broker would then be able to interpret the OWL-S markup to determine whether the service provides the capability it needs, to understand the input required to invoke the service, and to determine what information will be returned. This is accomplished in the OWL-S ontology through classes that describe what the service does (service profile), how to ask for the service, what happens when the service is carried out (service grounding), and how the service can be accessed (service model) [12].

III. Integrated METOC Broker

Our work on the IWB is focused on automated integration of METOC Web Services. We are engineering the IWB to automatically discover METOC Web Services and dynamically translate data and methods across them. The IWB's Web Service search and discovery function is illustrated in Figure 1. We are developing the IWB to search identified registries for METOC Web Services using the search feature supplied by that registry.



.

Fig. 2. The IWB search and discovery function

The IWB's mediation function is depicted in Figure 2. We are developoing it to dynamically translate user requests to differing Web Service interface specifications. For example, this shall assist with brokering requests to multiple METOC data providers whose services may have implemented a) a community standard interface such as JMBL, b) an interface that is not a DoD community standard (such as may be found among U.S. coalition partners), or c) an evolving version of a community standard interface.

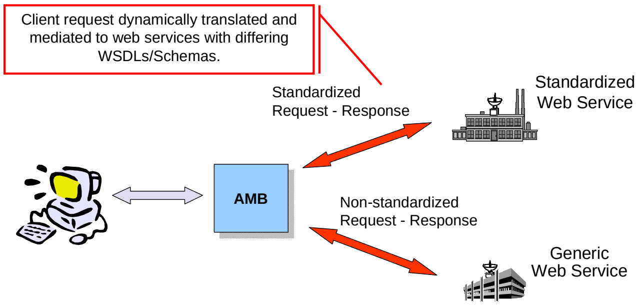

Client request dynamically translated and mediated to web services with differing WSDLs/Schemas.
Standardized Request - Response
Standardized Web Service
AMB
Non-standardized Request - Response
Generic Web Service

Fig. 2. The IWB mediation function.

While we are investigating the use of domain ontologies to automate the IWB, some of the IWB tasks seem suitable for resolution by automated classification techniques. One benefit of these techniques is that they do not require formal domain definition. More importantly, an automated classification approach does not rely on a data provider's deployment of additional specialized ontological descriptions of their Web Service, which is often lacking.

Identifying whether a particular Web Service supplies METOC data can be framed as a classification task, which involves assigning one or more predefined labels to an unlabelled object. Thus, the Web Service identification task involves assigning the label "METOC" or "Non-METOC" to a given Web Service.

A. IWB High Level Architecture

We first describe the mediation of user requests for data. This step includes the transformation of user requests and Web Services responses. The steps involved are:

Receive an XML formatted user request for data.

Decompose the user request to identify those XML tags that have associated values.

Locate the tag that corresponds to a "parameter" synonym.

This tag identifies the data request using the end-user's terminology.

Query the ontology for the concept corresponding to the term provided by the user.

Query the Dynamic Knowledge Base by this concept to obtain all Web Services that provide data related to the concept.

Transform the user's request to target web service's request structure. Where the request must be brokered to multiple Web Services, there may be multiple transformations. This step utilizes the XML template recorded during the discovery process.

This is an example of an IWB request XML message for salinity data in the specified area of interest.

```
<GridRequest xmlns:xls=http://www.w3.org/2001 /XMLSchmea-instance>
            <Parameter>salinity</Parameter>
            <aoi westLon="-90" southLat="10"
                  eastLon="-80" northLat="20"/>
</GridRequest>
```

This is tnen transformed in IWB to become a complete Web Service request XML message. Aside from the restructuring note that term " salinity" in the original request has been converted to the term "sal" by use of the ontology.

```
<GridDataRequest xmlns:xls="urn:nrl:METOC">
            <param>sal<param/>
            <areaOfinterest>
                  <westLongitude>-80<westLongitude/>
                  <southLatitude>-80<southLatitude/>
                <eastLongitude>-70<eastLongitude/>
                <northLatitude>-80<northLatitude/>
            <areaOfinterest>
</GridDataRequest>
```

Next we will describe in some detail the overall architecture that integrates the IWB processes. The functional components of the IWB are shown in Figure 3a and 3b. The IWB can begin mediating user requests once its Mapper component has discovered Web Services and begun populating the Dynamic Knowledge Base. Specifically, the Mapper takes as *input* (1) discovered Web Services populating the Dynamic Knowledge Base. Specifically, the Mapper takes as *input* (1) discovered Web Services interface specifications and (2) the METOC ontology. It uses this information to build the Dynamic Knowledge Base, and it also assigns a qualitative and quantitative confidence score to each service.

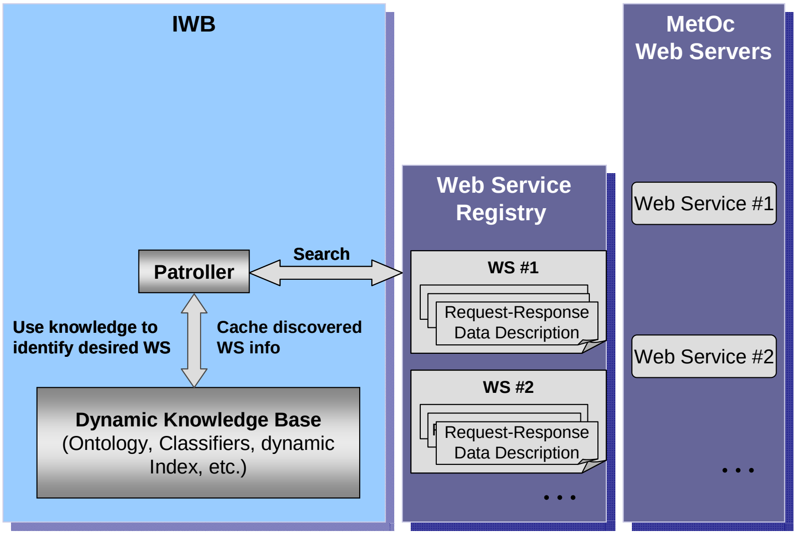


Figure 3a. IWB Architecture – Dynamic Discovery

After the IWB is initialized it is ready to process user requests to the appropriate web service or multiple services. The Mediator is the component of the IWB that provides the necessary transforms for this to occur. Clients submit data requests to the Mediator in an IWB XML format. The Mediator uses the previously created mappings to translate the client request into a candidate web service format\specified in the Web Services Registry and submits the request to the web service provider. As the recipient web service sends the data response back to the Mediator, the web service response is transformed by the Mediator to the end-user format and forwards it to the IWB's Client. This is the inverse of the request mapping process.

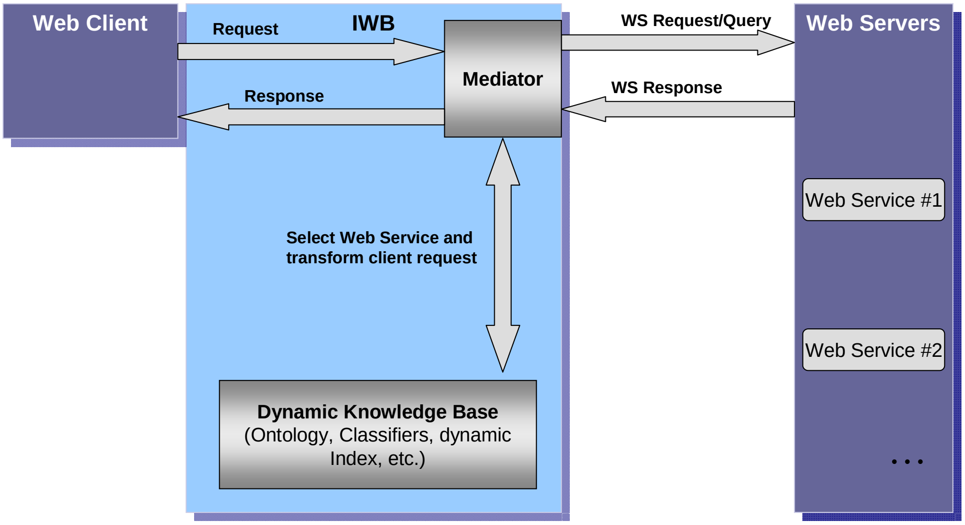


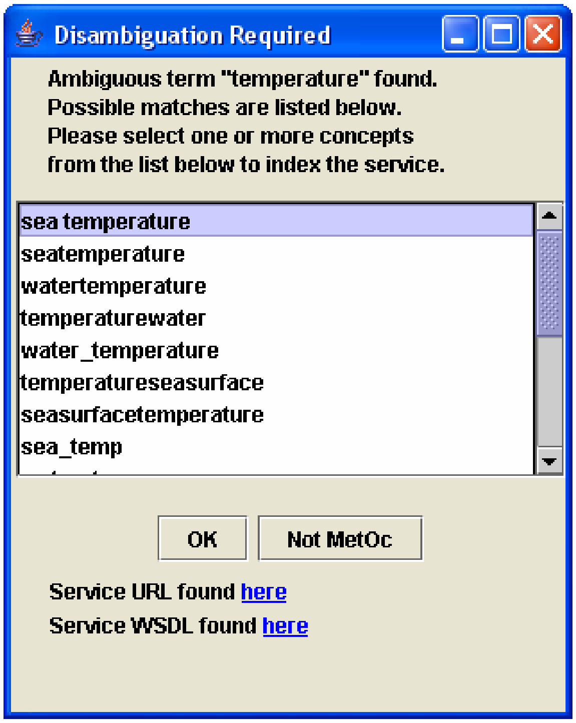
Figure 3b. IWB Architecture – Dynamic Mediation

The IWB performs two tasks: automated discovery and classification of web services that produce MetOc data, and syntax-independent consumption of this data by clients utilizing an ontology of domain information for identification of MetOc services. For instance, the ontology captures the top-level concept of a MetOc "Parameter." An instance of this class, such as "Sea Temperature" may have synonyms: "SeaTemp" and "TempSea". As a new web service is corralled by the IWB, its service description is broken into lexemes and matched to terms in the ontology. The ontology is manually constructed and maintained by domain experts, which results in a concise data model. However, small variations in a service description may thwart proper classification. For example, a service which offers a sea temperature parameter as "sTemp" may fail precise term matching, but there may be enough information to facilitate semi-automated ambiguity resolution. Another problem encountered while trying to index some web services was the non-uniformity of labeling and describing web services. For any given concept in the ontology, there could be many different synonyms that mean the same thing. Some services were labeled with terms that were similar to concepts in the ontology, but not exact matches. One example is the term "temperature." Using just the term, it is unclear whether the web service provides air temperature, sea temperature, surface temperature, etc.

The IWB therefore employs a partial matching system to insulate the classification from unnecessary failure which generates a similarity measure to be used in resolving ambiguous cases. Many such metrics exist, such as the Levenshtein edit distance. The N-gram distance proves to be a fast method that performs well in the types of variations present in MetOc web service descriptions. The IWB will then both index the service with a recording of the similarity value and utilize a GUI currently being implemented to allow expert user guidance in the disambiguation.

Specifically, terms from a web service that were not exact matches for concepts in the ontology are evaluated as partial matches. The list of possible partial matches is returned to the IWB and a disambiguation window is then displayed on the IWB server monitor. This allows the user in charge of maintaining the IWB server to select which concept to index the web service under. In order to assist the user in deciding which concept fits the web service in question, links to the web service and WSDL are provided. If it is determined that the service is not a MetOc web service, the user may click Not MetOc, and the service will not be indexed.

Due to the interest in use of Open Geospatial Consortium (OGC) Web Coverage Services (WCS) for these types of data, we have also extended the capability of the IWB to integrate data from WCS sites. WCS supports retrieval of geospatial data as "coverages" – that is, geospatial information representing space-varying phenomena. WCS structurally differs from World Wide Web Consortium (WC3) Web Services standards (e.g., WSDL) but does utilize formal XML structures to provide three operations: GetCapabilities, DescribeCoverage and GetCoverage. We have found these three sufficient to integrate a new data source. We have been able to effectively integrate a NATO Underwater Research Center (NURC) WCS into the IWB, including index and data retrieval. We are currently collaborating with NURC on the use of IWB for their data fusion center operations. It should be noted that future OGC plans are to provide a web service capability for WCS similar to WC3 standards, which would facilitate use of IWB.



Ambiguous term "temperature" found.
Possible matches are listed below.
Please select one or more concepts
from the list below to index the service.

sea temperature
seatemperature
watertemperature
temperaturewater
water_temperature
temperatureseasurface
seasurfacetemperature
sea_temp

Service URL found here
Service WSDL found here

Figure 4 - Sample Disambiguation Window for Term "temperature"

IV. Case-Based Classification

Case-based classification proceeds as follows. To classify a new object, it reuses the classifications of previously classified objects (i.e., *cases*) that have characteristics similar to the new object [13]. For example, each object in a table is a case and the list of objects in the table constitute the case base [14,15]. To assess the similarity of one case with another, the classifier uses a similarity metric. For example, the well known Euclidean distance metric can be used as a similarity function. The cases that are the most similar to the unclassified object are called its nearest neighbors. The classifier considers the classes of the *k* nearest neighbors from the case base when predicting the class label of an unclassified object. Training the classifier typically implies estimating the parameters of the similarity metric. Next, we describe the case-based approach we use for the Web Service classification task.

Web service classification in the IWB entails assigning one of two labels, “METOC” or “non-METOC”, to a Web Service in question. The input to the classifier is a Web Service schema described using the WSDL [16] and the output is an associated label. The process of training the classifier on example cases is shown in Figure 5.

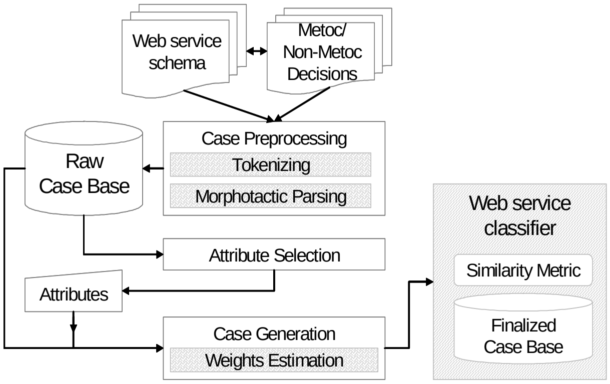


Fig. 5. Web Service classifier training process.

B. Case Pre-Processing

The WSDL describes the messages accepted by a Web Service and either contains or references an XML Schema. For classification, each WSDL must be converted into a case with attributes and values. We treat all the element contents in the associated schema as a source of attributes. For example, an element in a schema may contain the enumerated value “waterTemperature”. Its content can be directly used as an attribute. Alternatively, to reduce the sparseness of cases, it can be decomposed into its constituent terms. This is performed by a tokenization process, which decomposes such a string into its constituent words. For example, “waterTemperature” is decomposed into “water” and “temperature”. Subsequently, a morphotactic parsing process further reduces words into their baseforms [17]. For example, the word “producer” is reduced to its baseform “produce”. This approach allows us to reduce a Web Service schema to a bag of unique baseforms. Each baseform is a potential case attribute, where the frequency of its occurrence in a particular schema is its value. This is stored as a raw case in a preliminary case base. For each case, the decision of whether it is “METOC” or “non-METOC” is added as its class attribute.

C. Attribute Selection

With potentially hundreds of example Web Services for classifier training, we expect to generate thousands of attributes. This poses a serious computational challenge to the classifier and can also adversely affect classification performance by introducing noisy and irrelevant attributes. For example, the attribute “http” may appear in all cases and provide no useful information to discriminate METOC from non-METOC Web Services. To counter this problem, we perform attribute selection, where a metric is used to select a subset of attributes with a potential to improve classification performance. Numerous attribute selection metrics exists, including mutual information, information gain, document frequency [18], and rough set methods [13]. We apply the information gain metric to select attributes in the Web Service Classifier.

After the attributes have been selected, each case must be indexed with the selected attributes and their corresponding weights must be computed. In this initial study, we use the information gain metric to calculate the weights applicable to the attributes. This results in a classifier that includes the finalized cases and the similarity metric.

D. Case Generation

After training is complete, the classification of a previously unknown Web Service proceeds as follows. A web service whose classification is unknown is submitted to the classifier. Case pre-processing and case generation processes are used to convert the Web Service schema into a case. This case is matched with the cases in the case base using the learned similarity metric and its k-nearest neighbors are retrieved. Their classes are then applied to the new case as follows. Each nearest neighbor votes on the decisions based on its classification. Each vote is weighted by the similarity of the voting neighbor. The classification label with the most (weighted) votes is assigned as the class of the new case. If the class assigned to the new case is the same as its actual class, then this is counted as a correct classification. Classifier performance is measured by the percentage of cases classified correctly.

E. Evaluation

We evaluated the Web Service Classifier. For our study, we implemented the classifier’s preprocessor, attribute selector, and case generator. We obtained a set of 64 Web Services schemas from registries on the Web. Our meteorological subject matter expert then classified 26 of these schemas as METOC relevant. We used a leave-one-out cross-validation (LOOCV) method to evaluate our classifier’s performance, in which we repeatedly remove one case from the data set for testing and use the remaining cases to train the classifier. The classification accuracy for each test case is recorded using their respective trained classifier. This process of training and classification is repeated for each case in the set to determine the classifier’s average classification accuracy.

The maximum classification accuracy of the Web Service Classifier was 93.75%, at k=5 and the number of attributes = 523 (out of maximum possible 1790). We used a genetic algorithm to search for the values of the parameters k and the number of attributes threshold used in the information gain feature selection algorithm. We used classification accuracy as the fitness function for the genetic algorithm

V. Conclusion

We described a novel method of automating the identification of METOC Web Services within the context of an intelligent broker, the IWB. In this context, we described a case-based classification approach for Web Service identification. We reported the accuracy level achieved by our approach. In addition to autonomously identifying METOC Web Services, the IWB will also be expected to independently match the user’s data request to the correct method within the web service, to translate the user’s request to the Web Service request, to dynamically invoke the method on the service, and to translate the Web Service response. These issues are more complex than Web Service identification. Whether classification approaches may prove beneficial in addressing these tasks is a focus of our future research. Additionally, as part of its mediation function, the IWB may also have to invoke multiple Web Services where the data required by the user is not readily available from a single service. Also significant to the end-user is the IWB's assessment of data confidence and reliability. We believe that current findings warrant additional work on the applicability of classification approaches to automating machine discovery and integration of Web Services.

ACKNOWLEDGMENT

The authors would like to thank the Naval Research Laboratory's Base Program, Program Element No. 0602435N for sponsoring this research.

. REFERENCES

[1] Director, Force Transformation, Office of the Secretary of Defense, "Network-Centric Warfare Creating a Decisive Warfighting Advantage", Washington, DC , Winter 2003. Cleared for Public Release by Department of Defense directorate for Freedom of Information and Security Review 04-S-0272.

[2] Paolucci, M., Soudry, J., Srinivasan, N., & Sycara, K., (2004), A Broker for OWL-S Web services, *Proceedings of the AAAI Spring Symposium on Semantic Web Services*.

[3] Andreas Heß and Nicholas Kushmerick, (2003) Automatically attaching semantic metadata to Web services, *Proceedings of IJCAI*.

[4] Andreas Heß and Nicholas Kushmerick, (2004) Machine Learning for Annotating Semantic Web Services, *Proceedings of the AAAI Spring Symposium Semantic Web Services.*

[5] XML Schema Tutorial, 2004, http://www.w3schools.com/schema/schema_intro.asp.

[6] Web Services Definition Language, 2004, http://www.perfectxml.com/WebSvc3.asp.

[7] Cerami, E., 2002, *Web Services Essentials*, O'Reilly and Associates, 2002.

[8] http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/

[9] http://www.w3.org/TR/2004/REC-owl-features-20040210/

[10] http://www.aifb.uni-karlsruhe.de /WBS/cte/ontologyengineering/

[11] http://protege.stanford.edu/overview/

[12] http://www.daml.org/services/owl-s/1.1/overview/

[13] Cain, T., Pazzani, M.J., & Silverstein, G. (1991). Using domain knowledge to influence similarity judgment. *Proceedings of the Case-Based Reasoning Workshop* (pp. 191--202). Washington, DC: Morgan Kaufmann.

[14] Gupta, K.M, Moore, P.G., Aha, D.W, & Pal, S.K. (2005). Rough-Set Feature Selection Methods for Case-Based Categorization of Text Documents, in S. K. Pal, S. Bandyopadhyay, & S. Biswas (Eds*.) Lecture Notes in Computer Science (LNCS 3776)*, (pp. 792-798) : Heidelberg,Germany: Springer.

[15] Aha, D.W. (1992). Generalizing from Case Studies: A Case Study, in D. Sleeman & P. Edwards (Eds.), *Proceedings of the Ninth International Workshop on Machine Learning (ML92*) (pp. 1-10) San Mateo, CA: Morgan Kaufmann.

[16] WSDL (2005), Web Service Description Language, Version 1.1, http://www.w3.org/TR/wsdl.

[17] Gupta, K.M., & Aha, D.W. (2004). RuMoP: A Morphotactic Parser, in R. Sangal & S.M. Bendre (Eds.), *Proceedings of the International Conference on Natural Language Processing (ICON-2004*), (pp. 280-284). Mumbai, India: Allied Publishers.

[19] Yang, Y., & Pederson, J. (1997). A comparative study of feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412-420). Nashville, TN: Morgan Kaufmann.

[20] Open Geospatial Consortium (2006) Web Coverage Service (WCS) Implementation Specification, OGC 06-083r8, Version 1.1.0

[21] Department of Defense – JMBL (2006) http:// www.cffc.navy.mil/METOC/.