

Extended Semantic Web Services Model for Automatic Integrated Framework

Okkyung Choi, Sangyong Han and Ajith Abraham

Department of Computer Science & Engineering, Chung Ang Univ. Seoul, Korea
okchoi@ec.cse.cau.ac.kr, hansy@cau.ac.kr, ajith.abraham@ieee.org

Abstract

Semantic web services enable discovery, execution and composition of automated web services by combining web services based on standards, such as SOAP, WSDL and UDDI, with semantic web technologies such as RDF, DAML+OIL and OWL. In this paper, problems of the existing web services are analyzed and the extended semantic web service model applying the semantic web technology is suggested as a solution to these problems. The suggested model can perceive the service user's requests accurately by using the ontology server and provides automated and integrated framework based semantic web services through priority ranking of search results using the matchmaking service and rule-based service.

1. Introduction

The most desired future World Wide Web would consist of semantic services providing more accurate and reliable information and web services that provide users with improved high-quality service by automated methods instead of only providing simple information. Such technologies pursue a conversion to decentralized services that support heterogeneous environments.

The web services here refer to software systems that help interaction between heterogeneous platforms by structuralizing data using XML and sending the structuralized data through existing web technologies such as HTTP. Web services technology provides RPC mechanisms of the Web's decentralized environment based on the XML based standards such as SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration) etc. It provides a ground for businesses to open their business logics to the public and enables B2B (Business-to-Business Integration), that is, efficient to provide integration between businesses.

Web services are one of the key technologies in e-business and presently research and development of languages for constructing semantic web services, such as DAML-S, WSPL, X-LANG and BPEL4WS, are

underway in various fields. As for DAML-S, a method for accessing the existing web services method from the semantic web environment, weak points of former methods have been improved to enable effective web services registration, search, organization, execution and composition.

However, the current semantic web services model DAML-S also has some disadvantages in supporting automated web services. First, the model does not use the appropriate method for expressing the information, in short, the restriction conditions, and the user's requests are not applied sufficiently. Second, measurements of the service quality (fusibility, integration, performance, security and reliability) are inadequate. For such reasons, this paper suggests the extended semantic web services model to solve the above problems and enable efficient web services search and construction. The suggested model is different from the existing model in three aspects. First, it provides domain semantic information, which uses the DAML-S ontology, in order to reflect the user's requests accurately. Second, the matchmaking engine provides efficient matching of service requesters and providers and priority ranking of search results. Third and last, the search results produced by the matchmaking engine and the user's rules obtained from the user profile registry are compared in order to provide the service requester with the most appropriate information.

This paper is organized in the following order. First in Section 2, the existing semantic web services system is compared and analyzed with the suggested model. In Section 3, the matchmaking engine is described and in Section 4, the newly improved extended semantic web services model is suggested. Finally in Section 5, conclusions will be made along with plans for further studies.

2. Related Work

InfoSleth [2] is an agent based information search system, which uses the "broker agent" in order to

execute syntactic and semantic matchmaking. The broker agent supports connections between service providing agents and holds recent information in the repository and the broker provides query agents, which locate all possible agents, in order to provide the appropriate service. The specific methods used here are syntactic brokering and semantic brokering. Meanwhile, InfoSleth carries out services applying rules of LDL++ [13], logical deduction language. This language is not taken into account in this study since it is not a standardized service description language or a design method based on semantic web services.

In [4], a semantic web services system using OWL-S based brokers, uses agent based brokers for efficient web services execution and synchronization is presented. In this research, a new OWL-S is suggested in order to clarify the broker's functionality. The broker architecture shown in Figure 1 uses the Query processor and Discovery engine in order to provide service requesters with efficient and accurate service advertisement. However, no specific design plans are suggested nor any descriptions of matchmaking and brokering. Instead, descriptions mainly cover only the theoretical factors such as abstraction and pruning algorithms.

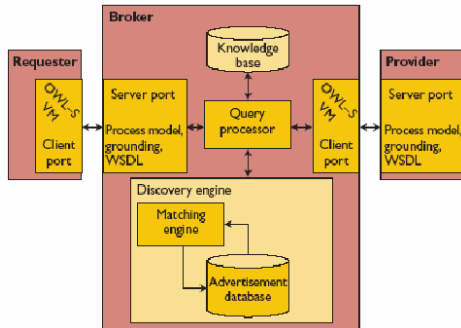


Figure 1. The broker architecture

As described in [10], which is composed of DAML-S based on semantic web services description, the standard for matching web services search results is whether the service requester's request and the service provider's advertisement match or not. The matches are made by comparing each of the service inputs and outputs and classified into four matching levels: Exact, Plug-In, Subsume and Fail. This service grading method using the matching algorithm enables efficient web service searching but not capable of drawing accurate service search results because the matching results of each level are not ranked in detail.

3. Matchmaking engine

3.1 The Definition of Matchmaking and Requests

Matchmaking is a process of finding the service provider that satisfies the server requester's requests. Matchmaking is executed based on whether the web service request and web service advertisement match or not. The match between requests and advertisements is determined based on whether the service input and output among the functional description match or not. The matchmaking system must support input and output through the repository and enable service browsing, correction and cancellation.

3.2 Matchmaking Algorithm

For efficient semantic web service searching, matching service requests and service advertisements must be done accurately. The match between requests and advertisements is made based on the match between inputs and outputs of the functional description. In other words, when the factors of the service request input and the service advertisement input match each other, the two inputs match, and when factors of the service request output and factors of the service advertisement output match each other, the two outputs match. As so, when all inputs and outputs match, the service executes the service request appropriately and provides satisfying results.

- [Rule 1] Exact
If advertisement A and request R are equivalent concepts, we call the match Exact. ($R = A$)
- [Rule 2] PlugIn
If request R is super-concept of advertisement A, we call the match PlugIn. ($R \sqsupseteq A$)
- [Rule 3] Subsume
If request R is sub-concept of advertisement A, we call the match Subsume. ($R \sqsubseteq A$)
- [Rule 4] Intersection
If the intersection of advertisement A and request R is satisfiable, we call the match Intersection ($R \cap A$)
- [Rule 5] Fail
If advertisement A and request R are not equivalent concepts, we call the match Fail. ($R \neq A$)

In this research, whether the input and output match or not is judged by classifying the matches into five different levels: Exact, PlugIn, Subsume, Intersection and Fail. As the level goes up from [Rule 1] to [Rule 5], the ranking is lower.

In this case, confusion could occur between [Rule 2] and [Rule 3]. For example, when a service giving "man" type outputs is desired, but the "human" type output was given instead due to the fact that "man" is a subclass of the "human" class, the service provided

would not be appropriate. As so, in the present research, unlike the matching methods of [10, 11], when a service receiving inputs of “man” based on [Rule 2] and [Rule 3] is desired, priority ranks are assigned when the request input value is the same or larger than the advertisement input value. The match ranking method applied in this research is largely divided into Steps 1 and 2. In Step 2, a new ranking algorithm [7], a modification of the former vector model, is applied for [Rule 2] and [Rule 3] to produce more detailed ranking. This newly suggested match ranking algorithm is described as follows:

Step 1

```

First_Match_Compare(output.R, output.A)
{
  if output.R is equivalent to output.A then Level =
  Exact; return Exact;
  else if output.R is SuperClassOf output.A then Level
  = PlugIn; return PlugIn;
  else if output.R is SubClassOf output.A then Level =
  Subsume; return Subsume;
  else if output.R is not incompatible with output.A
  then Level = InterSection; return InterSection;
  else Level= Fail; return Fail;
}
R:request, A:Advertisement

```

Step 2

```

Secound_Match_Compare(output.R, output.A)
{
  Switch(Level) {
    case Exact:
      Level_rank = 0;
      Break;
    case PlugIn: case Subsume:
      Call Function Ranking_Compare();
      Break;
  }
}

```

The two-step match ranking algorithm is applied in the case where the matching levels are Exact, PlugIn and Subsume. When the matching level is Exact, it means that the service request and the service advertisement are exactly the same and so this level is ranked at the highest match rank. In the case where the service request comprises the service advertisement, the Ranking_Compare() function dealt with in the former study Semantic Management Model [7] is called. The Ranking_Compare() function is indispensable to ranking the services within the same level. It uses the relationship, that is, the vertical and horizontal closeness, between succeeding levels and the synonym relation between terms to rank the matches. As so, a more detailed –two-step match ranking method

is produced to improve the former one-step simple match ranking method in order to provide clearer priority ranking of search results and more accurate and efficient search results.

3.3 Matchmaking Algorithm Application

3.3.1 Semantic Web Services Scenario

A customer wants to purchase a cellular phone through web service. The following describes the processing results after web service request. Mr. Park, a car salesman, wants to purchase a camera built in mobile phone made by Samsung so he can show the newest car models to his customers. But if the Samsung phones cost over 200 thousand won he considers purchasing an OEM brand product instead. Mr. Park used the web service to buy a cellular phone that satisfies such conditions. Here, the extended semantic web services system’s web agent obtained information from Mr. Park’s personal information profile that he prefers Hanmac phones among OEM brand products and within the same cost range phones with more functions. Based on this information, Hanmac’s camera built in cellular phone costing below 200 thousand won was recommended to Mr. Park. The web service even executes the purchase process if Mr. Park decides to buy the recommended product.

3.3.2 Definition of Ontology

This section suggests the definition of the ontology based on the scenario given in Section 3.4.1. In order to define the ontology, the advertisement and request services were defined first applying the DAML-S 0.9 version services profile class. In addition, the syntax rules were expressed using DL notions of the DAML+OIL syntax and the expression method is as shown below.

$$\begin{aligned}
 &ServiceProfile \sqsubseteq \perp \\
 &Advertisement \sqsubseteq ServiceProfile \\
 &Query \sqsubseteq ServiceProfile \\
 &Sales \sqsubseteq (=| providedBy.Provider) \cap \\
 &\quad (=| hasFeatureSelection.FeatureTyper) \cap \\
 &\quad (=| hasQuantity.Integer) \cap \\
 &\quad (=| hasProductPrice.Integer) \\
 &Provider \sqsubseteq (=| hasName.ProviderName) \cap \\
 &\quad (=| hasCompanyNmae.CompanyName) \\
 &Product \equiv hasBuilt-inCamera.Item \\
 &Item \equiv Cellular-Phone \cup PDA
 \end{aligned}$$

3.4 Matchmaking Engine Structure

Figure 2 illustrates how the service advertisement the service requester has requested is searched for using

the matchmaking engine. The procedures for each step are described below.

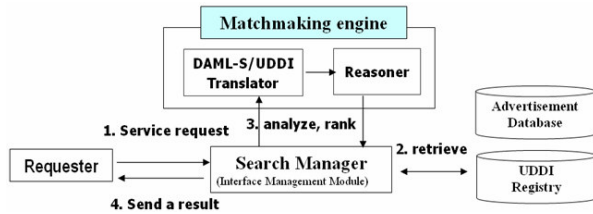


Figure 2. Matchmaking Engine Structure

Step 1: Requester submits a request to the Search Manager.

Step 2: Search Manager retrieves all the advertisements from the Database and UDDI Registry.

Step 3: Search Manager sends it to DAML-S/UDDI Translator that constructs a UDDI service description. Then Reasoner which is residing in the DAML-S/UDDI Translator analyzes, computes the level of match and ranks the results.

Step 4: Search Manager sends results to the requester. All the matched advertisements will be displayed as a result.

4. Extended Semantic Web Services Model

4.1 Problem Analysis

Based on web services related quality evaluation factors, this section will deal with the problems with web services in the semantic web environment. The problems can be described largely in four aspects as outlined below.

(1) Processing User's Request

The basic web services methods including UDDI are simple search methods such as syntax analysis that do not apply semantic factors. Furthermore, services, such as the artificial intelligent agent, which describes the information desired by the user and compare the search results are not provided. As a result, it is difficult to verify and confirm the user's requests.

(2) Composition of Services

For integration and composition between unit services, separate ontology information must first be integrated. However the current web services framework does not take care of this sufficiently. Web composition languages used at present include XLANG suggested by Microsoft, WSFL (Web Service Flow Language) by IBM and BPEL4WS (Business Process Execution Language for Web Services) suggested by Microsoft and IBM together. However, because these languages are all different standardization processes,

they rather function as obstacles to environment integrating factors. Furthermore, there is a need for an automated web services composition technology the same as the agent technology.

(3) Service Quality Measurement

The most important factor of the semantic web service is how much the user can trust and rely on the service provided. However, other web services methods currently used do not process reliable messages and the waiting time, message processing and transferring time is too long resulting in poor performance.

(4) Satisfaction with Search Results

The current web services search methods do not apply priority ranking of search results or automated classification methods. A ranking method that considers users' requests and preferences is greatly required.

4.2 Solutions

(1) User's Request Processing

Problems with the simple searching method can be solved by the E-engine Ontology Server [7]. In order to apply semantic factors, search for groups of analogous terms is requested and the ontology system returns semantic information of the search results.[7] Additionally, the user's request can be verified and applied by using the Search Manager [7] suggested in former studies and executing repeated queries for analogous terms, and the current UDDI's simple search method can be improved to enable efficient matching between service requests and advertisements by using the matchmaking engine described in Section 3.4.

(2) Composition of Services

For interoperability between separately composed unit services, the information required for composition must be provided between each unit system and automatic execution should be carried out between each service. As an example, before going on a business trip, reservations for plane tickets, hotel rooms and rental cars must be made separately but these unit services must at the same time be well organized to form a good composition. For such purposes the extended DAML-S capable of rule based searches is suggested. Further descriptions of this will be given in Section 4.3.

(3) Service Quality Measurement

In order to provide reliable services, an automated semantic web services method is required. However, the DAML-S that currently supports this function does not include rule information, and so the user's restrictions are not applied accurately. In order to solve such problems with DAML-S, this study uses

DamlRuleML [6], the DAML-OIL ontology language for RuleML, to include rule information in DAML-S and produce an extended DAML-S improved model. Since DamlRuleML is based on the grammar of DAML+OIL, the matter of compatibility with DAML-S specifications shall not be a problem here.

(4) Satisfaction with Search Results

The semantic management module suggested in a former study [7] is used for providing priority ranked search results. The ranking of service results is supported by using the newly suggested similarity measurement model [7].

4.3 System Model and Functions

4.3.1 System Model

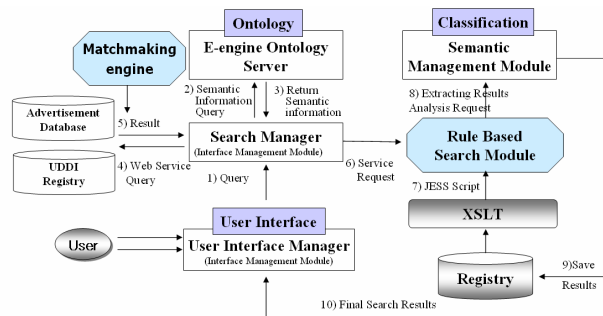


Figure 3. System Architecture

Figure 3 depicts the system model's general flow chart indicating the applied methodologies. It shows the whole procedure from the user request query to the results provided to the user. The suggested system model executes web service searches along a 10-step procedure. The ten steps are described below.

Step 1: User Request Query

The user makes a search query through the user interface.

Step 2: Request Search for Analogous Terms Groups

Search for groups of analogous terms is requested based on the returned search results. The E-engine ontology server provides the user with ontology terms and descriptions so that the user is able to select the right ontology.

Step 3: Production of Analogous Terms

A re-query is made in order to select the desired term among the search results of analogous terms groups. For example, food, groceries, cooking etc. are terms that would be placed in the group "foodstuffs" and terms such as original equipment manufacturing in the group "OEM." As so, the desired result can be drawn by the user's re-query.

Step 4,5: UDDI Search and Returning Search Results

The search for the user's search query is executed by using UDDI and the advertisement database (advertisement registry). Then the returned search results are analyzed using the matchmaking engine and the results are sent back.

Step 6: Request for Rule Based Search

Rule based search is requested based on the produced results.

Step 7: Execution of Rule Based Search

The results produced in step 4) and the user's rules acquired from the user profile registry are compared to select only the appropriate information. Going back to the scenario given in Section 3.4.1, the information from the user information profile that Mr. Park prefers Hanmac products among OEM products and Mr. Park's query for products under 200 thousand won are compared to provide Mr. Park with the right information. In this research, SweetJess [5] is used for executing rule based searches. SweetJess executes reasoning after expressing rules with DAML+OIL and converting them into JESS's rule language. With SweetJess, the rules indicated in the user information profile and the returned web service results are converted to Jess script through XSLT. Here the Jess engine [5] uses XSLT to convert rules expressed by RuleML to a format that can be used by the existing automatic reasoner and then draws the results through forward inference. [5]

Step 8: Request for Analysis of Extracted Results

For analysis of the extracted results, similarity measurement and ranking is done by requesting to the semantic management module, the classification processing module suggested in current studies.

Step 9,10: Return and Store Extracted Results

Even though the information of the service desired by the user is obtained through a rule based search, it is impossible to perceive whether the web service is currently valid and can be used by the user immediately. Due to this, the invalid services are extracted and ranked based on the user's search query through the similarity measurement model suggested in current studies [7]. Then the final results are returned and the information is stored in the registry.

4.3.2 Functions of Each Module

(1) User Interface Manager

The user interface manager provides the user search input screen and suggests the final results screen to the user.

(2) Search Manager

In order to enhance accuracy, the search manager brings the domain semantic information from the E-engine ontology server and draws accurate search

results through the user's query. Based on these results the user's query is requested to WWW and UDDI and the returned matching results are received.

(3) Matchmaking Engine

The overall matchmaking process is carried out by applying the matchmaking algorithm. The information obtained through the UDDI and advertisement registries are analyzed and the match results are ranked by the reasoner.

(4) E-engine Ontology Server

In order to semantically connect analogous terms, search for analogous terms groups using the ontology server is requested. In other words, if two terms are the same semantically although they may differ in structure, the terms are integrated into a single abstract schema or semantically connected.

(5) Rule Based Search Module

The module executes rule based searches to compare produced results with the user's rules drawn from the user profile registry and select only the appropriate service information.

(6) Semantic Management Module

The semantic management module automatically classifies and ranks the priority of the services selected by the rule based search module.

5. Conclusions and Future Studies

Recently the emergence of web services provided the foundations for businesses to open their business logics to the public efficiently. Web services have become the motive of B2Bi, that is, integration between businesses. The current web services system does not take the user's requests into account or provide sufficient evaluation of quality such as reliability and search result ranking.

As a solution to these problems, this paper suggests the extended semantic web service model, a fusion of the semantic web and web services technology. The suggested model enables efficient matches between service requestors and service providers with the matchmaking engine and also allows verification and application of the user's requests by the rule based search service through DAML-S expansion. For future research, a performance measurement algorithm must be designed for testing the efficiency and accuracy of the suggested model and comparison, analysis and evaluation shall be carried out based on the performance measurements.

6. References

[1] Daniel A.Menasce, "QoS Issues in Web Services", IEEE Internet Computing, 2002.12

- [2] M. Nodine, W. Bohrer, and A. Ngu. Semantic multibrokering over dynamic heterogeneous data sources in infosleuth. In Proc. of the International Conference on Data Engineering, 1999.
- [3] I. Horrocks, U. Sattler, and S. Tobies. "Practical reasoning for expressive description logics. In H. Ganzinger", D. McAllester, and A. Voronkov, editors, LPAR'99, pp. 161-180., Springer-Verlag, 1999.
- [4] C. Zaniolo. The logical data language (ldl): An integrated approach to logic and databases. MCC Technical Report STP-LD-328-91, 1991.
- [5] Benjamin N. Grosz 1, Mahesh D. Gandhe 2, Timothy W. Finin 2, "SweetJess: Translating DamlRuleML to Jess", <http://userpages.umbc.edu/~mgandh1/publication/SweetJess.htm>
- [6] [6]DamlRuleML, <http://userpages.umbc.edu/~mgandh1/DamlRuleML.html>
- [7] Okkyung Choi, Sangyong Han, "SW-IQS: Semantic Web based Information Query System for the integration of semantic data", Presentation of Studies at the Korea Information Processing Society Fall Edition, 2004
- [8] Massimo Paolucci, Julien Soudry, Naveen Srinivasan and Katia Sycara, "A Broker for OWL-S Web services", The Robotics Institute, Carnegie Mellon University
- [9] S. McIlraith, T.C. Son and H. Zeng, Semantic Web Services, IEEE Intelligent Systems, Special Issue on the Semantic Web, Volume 16, No. 2, pp. 46-53, March/April, 2001
- [10] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, Semantic matching of web services capabilities. In Proc. Of the 1st International Semantic Web Conference (ISWC), 2002.
- [11] Lei Li and Ian Horrocks, "A Software Framework For Matchmaking Based on Semantic Web Technology", In Proceedings International WWW Conference, Budapest, Hungary. 2003
- [12] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, S. A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. Daml-s: Semantic markup for web services. In Proc. of the International Semantic Web Workshop, 2001.
- [13] LDL++ Java Interface, <http://pike.cs.ucla.edu/ldl/>
- [14] David Trastour, Claudio Bartolini and Javier Gonzalez-Castillo, "A Semantic Web Approach to Service Description for Matchmaking of Services", HP Labs Technical Report.
- [15] Javier Gonzalez-Castillo, David Trastour, Claudio Bartolini, Description Logics for Matchmaking of Services, HP Labs Technical Report.