

The EEE-05 Challenge: A New Web Service Discovery and Composition Competition

M. Brian Blake
Georgetown University
Washington, DC, USA
blakeb@cs.georgetown.edu

Kwok Ching Tsui
The Hong Kong and
Shanghai Banking
Corporation, Hong Kong
kwokchingsui@hsbc.com.hk

Andrea Wombacher
University of Twente
Enschede, The Netherlands
a.wombacher@utwente.nl

Abstract

With growing acceptance of service-oriented computing, an emerging area of research is the investigation of technologies that will enable the discovery and composition of web services. Using the same approach as the popular Trading Agent Competitions (TAC), the EEE-05 Web Services Challenge is the first event geared towards the management of web services. The competition solicits industry and academic researchers that develop software components and/or intelligent agents that have the ability to discover pertinent web services and also compose them to create higher-level capabilities. This paper describes the competition details for this first year and expectations for future events.

1. Introduction

The purpose of the EEE-05 Challenge is to establish a venue where researchers can collaborate on implementations in the web service composition domain. In addition, the results from the competition can be used as performance baselines for other researchers. Furthermore, software design approaches can be demonstrated, enhanced, and disseminated each year as the competition evolves.

The objective of the competition, in the first year, is to encourage participants to concentrate on syntactical matching and chaining for Web Service Description Language (WSDL) documents. A successful software entry will be able to accurately and efficiently find services using the WSDL part names underlying input and output messages. Secondly, entry software is required to create chains of services by linking output part names to the subsequent input part names. The intent of the first year is for participants to create part name matching components/agents. The software created in the first year will set the foundation for later years of this competition (i.e. each year with more technical rigor). Each year, software entries should become more efficient.

2. The First Year

In this first year of competition, the web services repository will be based on WSDL 1.1. The participants were provided with two tutorial sites [2][3]. The organizers also suggested the use of the Microsoft .Net IDE as an editor for WSDL documents. The following sections describe the samples repository, discovery sample, and composition sample.

2.1 Samples Repository and Input Files

The samples repository can be found at [4]. The 2005 competition concentrates only on messages (and their underlying part names) and ports. Concrete descriptions such as bindings and services will be the focus in later years of the competition.

The service repository contains over 100 services. Many services have logical part names, however other services may be auto-generated with part names with the random combination of letters. Other services are sub-sets and variations of the *correct* services.

Participants will be provided with an XML file to initiate the discovery and composition routines in the competition. A sample of the XML request is shown in Table 1. In this first year, organizers will be flexible in allowing the competitors to reformat the XML file to best meet the participants' front-ends.

Software entries are required to execute on the designated competition workstation. In the first year, a Windows-based machine will be used, and participants are required to e-mail their system requirements prior to the competition.

Table 1. Sample Competition Input File.

```
<EEE05Challenge>
  <DiscoveryRoutine>
    <Provided> partname1, partname2 </Provided>
    <Resultant> partname1 </Resultant>
  </DiscoveryRoutine>
  <CompositionRoutine>
    <Provided> partname1, partname2 </Provided>
    <Resultant> partname1, partname2 </Resultant>
  </CompositionRoutine>
</EEE05Challenge>
```

2.2 An Example Discovery Routine

Participants will be asked to find a specific service that can fulfill a certain input and output criteria. Table 2 is a sample request file.

Table 2. Sample Discovery Request.

```
<DiscoveryRoutine>
  <Provided> foodPref, custStreetAddress , custCityAddress,
             custStateAddress, custZipAddress </Provided>
  <Resultant> restaurantName, restaurantID </Resultant>
</DiscoveryRoutine>
```

The most relevant service in the repository is the *findCloseRestaurant* service as shown in Table 3. The *findCloseRestaurant* service may be considered a bit over-qualified for the requirements, but it does fulfill them. Only one service will accurately meet the requirements in the repository. However, other services that partially meet the requirement are also included in the repository. Only accurate matches will count. The discovery portion of the competition is used to evaluate the design and speed of the software entries to execute the matching.

Table 3. Snippet of *findCloseRestaurant*.

```
<message name="findCloseRestaurant_Request">
  <part name="custStreetAddress" type="xs:string"/>
  <part name="custCityAddress" type="xs:string"/>
  <part name="custStateAddress" type="xs:string"/>
  <part name="custZipAddress" type="xs:string"/>
  <part name="foodPref" type="xs:string"/>
</message>

<message name="findCloseRestaurant_Response">
  <part name="restaurantName" type="xs:string"/>
  <part name="restaurantID" type="xs:string"/>
  <part name="restaurantStreetAddress" type="xs:string"/>
  <part name="restaurantCityAddress" type="xs:string"/>
  <part name="restaurantStateAddress" type="xs:string"/>
  <part name="restaurantZipAddress" type="xs:string"/>
</message>
```

2.3 An Example Composition Routine

Participants will also be posed with a composition request (finding a specific sequence of services). The routine in Table 4 can be fulfilled using the sequence, *purchaseALT.wsdl->reserveRental.wsdl->reserveRoom.wsdl->createItinerary.wsdl*, as captured in [4].

Table 4. Sample Composition Request

```
<CompositionRoutine>
  <Provided> firstName, lastName, middleInitial,
             creditCardNum, creditCardExp, creditCardSecID, departCity,
             departState, destCity, destState, rentalPref , roomPref, hotelName
  </Provided>
  <Resultant> ItineraryURL </Resultant>
</CompositionRoutine>
```

The *reserveRental* and *reserveRoom* require the output of *purchaseALT*, while *createItinerary* requires information from *reserveRental*, *reserveRoom*, and *purchaseALT*. The competition will limit three services as predicates for a subsequent service, however, the software entries are advised to keep a running memory of all available information. Participants should note that the most effective software will be combine front-to-back and back-to-front processing, perhaps simultaneously. In addition to the aforementioned more complex routine, other routines will require just two or three services with direct, straight-forward compositions.

3. Evaluation and Future Competitions

The organizers are using this first year as a case to establish the most effective approach to evaluating the software. The initial evaluation approach will consist of a subjective score on the system design. Other aspects will be performance and accuracy. One idea is to allow the discovery and composition to proceed for a limited amount of time and count the accurate number of discoveries or compositions, respectively. In this first year, the development of the most effective evaluation criteria is on-going. The results of this development will be published in other post-conference forums.

In second year, the competition will require participants to match part names that are not syntactically the same. In subsequent years, semantic languages such as OWL-S will be required.

4. Acknowledgements

This competition has been greatly influenced by the support from and fruitful conversations with Dr. William K.W. Cheung of Hong Kong Baptist University, Dr. Eleni Stroulia of the University of Alberta, Dr. Terry Payne of the University of Southampton, and Dr. Norman Sadeh of Carnegie Mellon University.

5. References

- [1] The Trading Agent Competition (2005): <http://www.sics.se/tac/page.php?id=1>
- [2] Microsoft Corporation: Web Service Description Language Tutorial, (2005) <http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/wsdlexplained.asp>
- [3] WWW Schools: Web Service Description Language Tutorial (2005), <http://www.w3schools.com/wsd/default.asp>
- [4] EEE-05 Challenge Samples Repository (2005), http://cssun.georgetown.edu/~blakeb/EEE05/WSDL_Repos.zip