The Web Service Challenge 2008

1 The Web Service Challenge

Since 2005, the annual Web Service Challenge[1] (WS-Challenge, WSC) provides a platform for researchers in the area of web service composition to compare their systems and exchange experiences. It is co-located with the IEEE Conference on Electronic Commerce (CEC) and the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE).

During the last years, the web service challenge focused on optimizing the service discovery and composition process solely, using abstractions from real-world situations. The taxonomies of semantic concepts as well as the involved data formats were purely artificial. Starting with this year's competition, the challenge will approach more practical scenarios. Therefore, the 2008 data formats and the contest data itself will be based on real world schemas for service compositions, services, and ontologies.

2 The Challenge

In the competition, we adopt the idea of so-called Semantic Web Services that represent Web Services with a semantic description of the interface and its characteristics. The task is to find a composition of services that produces from a set of given input parameters a set of queried output parameters. The overall challenge procedure is as follows:

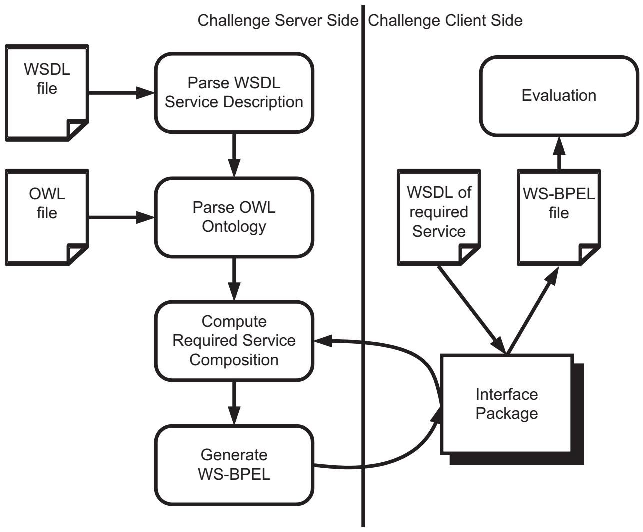

Figure 1: The procedure of the Web Service Challenge 2008.

The contestant's composer software is positioned on the server side and start with a bootstrap procedure. First of all, the system is provided with a path to a WSDL file. The WSDL file contains a set of services along with their input- and output parameters. The amount of services will change from one problem definition to the other. Every service will have an arbitrary amount of parameters. Then, we provide the address of the OWL file. This file contains the taxonomy of concepts used in this challenge according to the OWL schema. The bootstrapping process includes loading the relevant information from these files.

The challenge task will then be sent to the composer via a client-side GUI very similar to last year's challenge. After the bootstrap mechanism on the server side is finished, the GUI queries the composition system with the challenge problem definition. The contestant's software must now compute a solution – one or more service compositions

[1] see http://www.ws-challenge.org/ [accessed 2007-09-02]

– and answer in the solution format which is a subset of the WS-BPEL schema. When the WS-BPEL document is received by the GUI, we will stop a time measurement and afterwards evaluate the compositions themselves.

3 Evaluation

The Web Service Challenge awards the most efficient system and also the best architectural solution. The best architectural effort will be awarded according to the contestant's presentation and system features. The evaluation of efficiency consists of two parts as described below. The exact formula will be published soon.

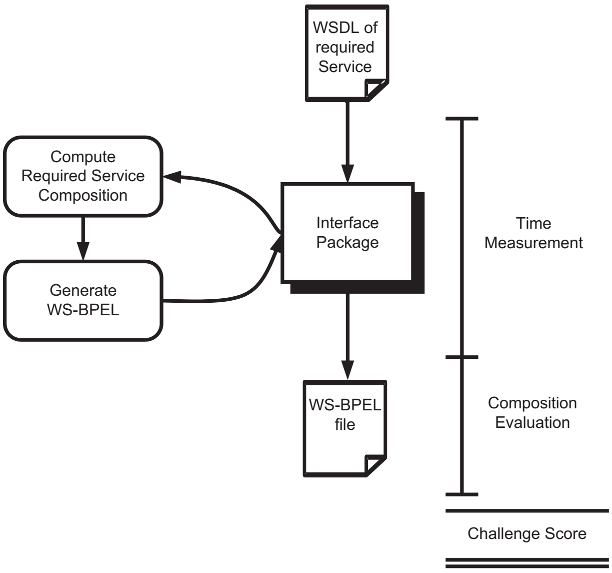

Figure 2: Evaluation in the Web Service Challenge 2008.

1. Time Measurement: The time measurement is done by the interface package. We take the time after submitting the query and the time when the composition result is fully received. The bootstrap mechanism is excluded from the assessment. There will be a time limit for bootstrapping after which a challenge is considered as failure.
2. Composition Evaluation:
   - Completeness: The amount of compositions discovered by the system.
   - Composition Length: The shortest composition.
   - Composition Efficiency: Parallel versus sequential execution of services.

4 What's New

Document Formats: OWL Ontologies, WSDL-Queries, WS-BPEL solutions schemas. Example files and documentation will follow shortly.
XSD-Types: The challenge will include matching XSD-Type definitions like arrays, simple types, complex types with substructures and enumerations.
Parallel Execution: The WS-BPEL schema supports the specification of parallel execution of services. Valid parallel execution will positively influence the systems challenge score.
No results: The challenge will include sets of services that actually deliver no solution. The system should act accordingly and cancel the discovery in time.
Images: We are currently investigating whether it makes sense to provide images for virtual machines along with the exact system specs of the systems that will be used to test the composers.