

A Batch Script Generator Web Service for Computational Portals

Stephen Mock
Kurt Mueller [1]

Choonhan Youn
Marlon Pierce
Geoffrey Fox [2]

Mary Thomas [3]

Abstract

Computational portal developers often re-implement functionality found in existing portals because there are no common discovery and access mechanisms in place to enable sharing of portal functions. The web services architecture provides an implementation independent, protocol based mechanism for entities to find, share, and invoke remote services. We developed complementary web services at SDSC and IU that generate batch scripts for different batch queuing schedulers.

1. Implementation

Two instances of the Batch Script Generator Web Service (BSG-WS) that generate queue scripts for the appropriate batch queuing systems were tested. Both SDSC and IU had previously implemented this service for existing portals (HotPage and Gateway), and wanted to develop an interoperable web service that could be used by existing projects.

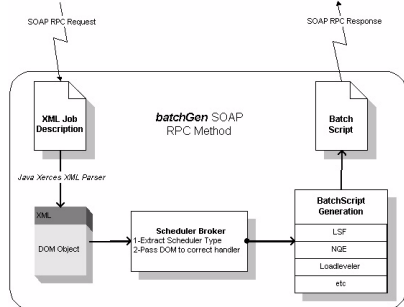


Figure 1: batchGen Web Service architecture.

The BSG-WS makes one method (batchGen) available for use by clients through a SOAP-encoded remote procedure call. The Java declaration of batchGen is

```
Public String batchGen(String xmlString)
```

This declaration establishes that batchGen takes a string, and returns another.

The input to the RPC method batchGen is a string of XML that describes the batch job for which the client wishes to create a batch script. The string output of batchGen is the batch script generated by the web service based on the XML job input, as detailed in Figure 1.

The WSDL file describing the BSG-WS was generated from the Java source code of a client of the BSG-WS using the WSTK [4].

Prototype web service entities that map directly to the portal service providers (Gateway and HotPage) were created in a UDDI repository [2]. For each such entity there exists one or more services (batch script generator, job submission, context management). Service bindings are used to point to SOAP web servers, the access points for our services. The WSDL interfaces are published in the UDDI directory, and links to these are placed within the service repository. Test cases were built that demonstrated that this system could be utilized by other web services, but we found that the batch script services could not be adequately described by UDDI due to the business oriented design of the UDDI and that modifications to the protocols will be required.

In conclusion, we find that WSDL and SOAP can be used to produce interoperable, language independent services, that can be shared across Grid domains and projects like the Gateway and HotPage projects.

2. References

- [1] San Diego Supercomputer Center, University of California, San Diego. {mock,kurt}@sdsc.edu
- [2] Community Grid Labs, Indiana University. {marpierc,cyoun,gcf}@Indiana.edu
- [3] Texas Advanced Computing Center, University of Texas at Austin. mthomas@tacc.utexas.edu
- [4] IBM Web Services Toolkit, available at <http://www.alphaworks.ibm.com/tech/webservices toolkit>.