A Web Service Composition and Deployment Framework for Scientific Workflows

Ilkay Altintas Efrat Jaeger Kai Lin Bertram Ludaescher Ashraf Memon
San Diego Supercomputer Center, University of California, San Diego
{altintas, efrat, klin, ludaesch, amemon}@sdsc.edu

Abstract

This poster presents the web services framework in the Kepler scientific workflow system and illustrates them with a real-world example.

1. Introduction

Kepler [1] is a system for the design and execution of scientific workflows. It is built on top of the Ptolemy II system, a modeling and design tool for assembling concurrent components by means of various models of computation [2].

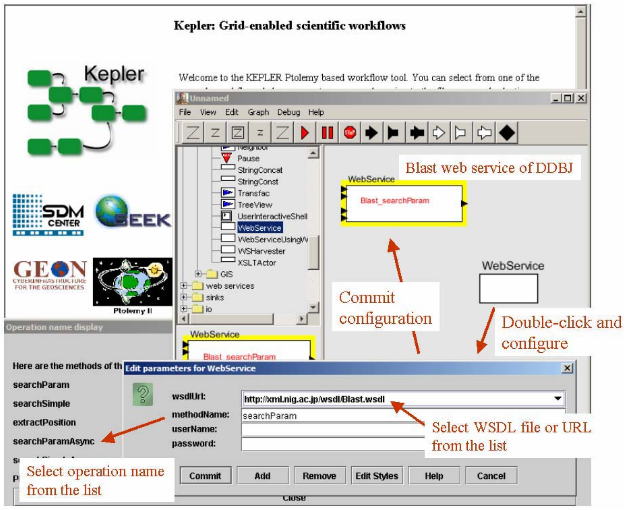
In a variety of disciplines (e.g., geology, chemistry, biology, ecology) scientists need adaptable interfaces and tools for accessing scientific data and executing complex analyses on the retrieved data. Such analyses can be modeled as *scientific workflows*. While traditional business workflows are oriented towards document processing, task management, and control-flow, scientific workflows typically are data-intensive and/or computationally expensive, dataflow-oriented, and often involve data transformations, analysis, and simulations. Kepler is unique in that it seamlessly combines high-level workflow design with execution and runtime interaction, access to local and remote data, and local and remote service invocation. Other unique features are inherited from the underlying Ptolemy II system, e.g., the ability to combine different models of computations in a clean way.

2. Web Service Components in Kepler

Kepler's web service components allow scientists to utilize computational web service resources and web service accessible data sources in a distributed scientific workflow. This functionality can be achieved by entering a single WSDL location or by specifying a web service repository (e.g. a UDDI repository or a web page with links to WSDL descriptions). For the latter, a search term can be specified so that only the matching web services are "harvested" and plugged into the system. Grid extensions to Kepler are described in [3].

2.1. Web Service Actor

Computational units in Kepler are called *actors*, which are reusable components that communicate with each other via input and output ports. We have implemented a generic web service actor that is used as a web service client in distributed workflows. Using this actor, any application that can be deployed as a web service, can be used as a Kepler actor.



Kepler: Grid-enabled scientific workflows

Welcome to the KEPLER Ptolemy based workflow tool. You can select from one of the

Here are the methods of th

wsdlUrl: http://xml.nig.ac.jp/wsdl/Blast.wsdl
methodName: searchParam
userName:
password:

Blast web service of DDBJ

Commit configuration

Double-click and configure

Select WSDL file or URL from the list

Select operation name from the list

Figure 1. Example instantiation of a web service actor

The WebService actor, as indicated in Figure 1, provides the user with a simple plug-in mechanism to execute any WSDL-defined web service. The user can instantiate the generic web service actor by providing the WSDL URL and choosing the desired web service operation. The actor then automatically specializes itself and adds ports with the inputs and outputs as described by the WSDL. The so instantiated actor acts as a proxy for the web service being executed and links to the other actors through its ports.

2.2. Web Service Harvester

Kepler provides a web service *harvester* capability for importing web services from a repository. This feature was developed for conveniently plugging in a whole set of (possibly related) services. The web service descriptions to be imported can be harvested from a simple web page or retrieved from a UDDI repository. Once imported, the web services are saved as actors. These actors can then be reused in different scientific workflows.

3. Deployment of Workflows as Web Services

The Kepler environment allows users to view and execute workflows on a web page using applets. This enables using the Kepler engine with no installation requirements. In order to use workflows execution results within other applications or execute compute intensive tasks on higher performance machines, Kepler also comes with a mechanism, to deploy workflows as web services.

Several obstacles are encountered while attempting to deploy a workflow as a web service. First, most of Kepler's output means use a separate display window. Second, no user interaction is available using web services.

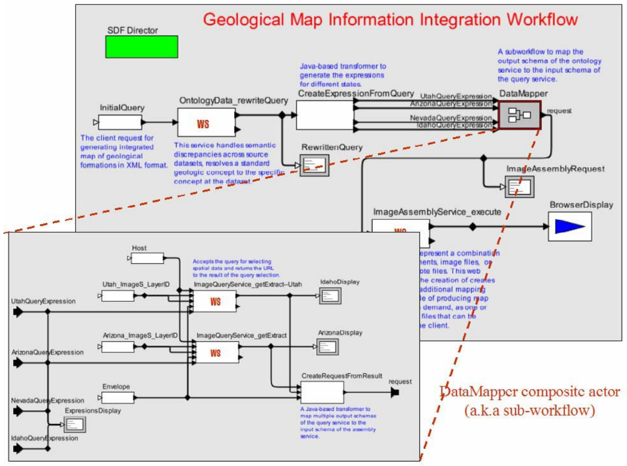
We overcome these obstacles using three different deployment models. The first model assumes no user interaction or separate display windows. The deployment of a workflow requires deployment of a small execution engine with the workflow. The second model assumes no user input. We provide Kepler with two execution modes: local and remote. In a remote mode all outputs are streamed into a file instead of a window channel. The mode is set either by the director or by each actor individually. The third model allows user input. A special interaction actor is added before each user interaction actor. This actor, when reached, returns the control to the client to provide the necessary input along with a session id pointing to the execution thread (the resume point).

4. Scientific Workflow Example

The current web service components of the Kepler system have been used in various scientific domains, including molecular biology, geosciences, and ecology.

One example is the "Geological Map Information Integration Workflow" depicted in Figure 2. This workflow was designed by a geologist to integrate State Geologic Maps using rock and geologic age ontologies. This model demonstrates the use of distributed processes within a workflow. See [4] for details of this workflow.

The application workflows show how to employ Kepler's web service components to compose distributed scientific workflows. Since web services are often not designed to fit, data transformations between the outputs of previous steps and inputs of subsequent steps are usually required. For this purpose, specialized data transformation actors (e.g. XSLT, XQuery) have been implemented. User interaction and workflow output are performed via a browser actor.



Geological Map Information Integration Workflow
SDF Director
Java-based transformer to generate the expressions for different states.
A subworkflow to map the output schema of the ontology service to the input schema of the query service.
InitialQuery
OntologyData_rewriteQuery
WS
CreateExpressionFromQuery
UtahQueryExpression
ArizonaQueryExpression
NevadaQueryExpression
IdahoQueryExpression
DataMapper
request
The client request for generating integrated map of geological formations in XML format.
This service handles semantic discrepancies across source datasets, resolves a standard geologic concept to the specific concept at the dataset.
RewrittenQuery
ImageAssemblyRequest
ImageAssemblyService_execute
BrowserDisplay
epresent a combination
nents, image files, or
ote files. This web
he creation of creates
additional mapping
le of producing map
demand, as one or
files that can be
e client.
Host
Accepts the query for selecting spatial data and returns the URL to the result of the query selection.
Utah_ImageS_LayerID
ImageQueryService_getExtract--Utah
IdahoDisplay
UtahQueryExpression
Arizona_ImageS_LayerID
ImageQueryService_getExtract
ArizonaDisplay
ArizonaQueryExpression
WS
CreateRequestFromResult
request
Envelope
NevadaQueryExpression
ExpresionsDisplay
A Java-based transformer to map multiple output schemas of the query service to the input schema of the assembly service.
IdahoQueryExpression
DataMapper composite actor
(a.k.a sub-workflow)

Figure 2. Geological Map Information Integration Workflow.

5. Future Plans

Based on our experiences with workflows from various application domains, new features are being added to Kepler. Currently, the web service actor operates only for basic XMLSchema datatypes. Research and planned development activities include extensions to the Ptolemy II type systems to allow for semantic types, optimization of web service quality, fault tolerance and fail over, and last not least recent Grid developments such as WSRF.

Acknowledgements. Kepler includes contributors from SEEK [5], SDM Center-SPA [6], Ptolemy II [2] and Geon [7]. Work supported by NSF ITRs 022567 (SEEK), 0225673 (GEON), DOE DE-FC02-01ER25486 (SciDAC/SDM), and DARPA F33615-00-C-1703 (Ptolemy).

6. References

[1] Kepler: An Extensible System for Scientific Workflows, http://kepler.ecoinformatics.org

[2] Ptolemy II, http://ptolemy.eecs.berkeley.edu/ptolemyII

[3] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock, "Kepler: Towards a Grid-Enabled System for Scientific Workflows", in the Workflow in Grid Systems Workshop in GGF10 - The Tenth Global Grid Forum, Berlin, Germany, March 2004.

[4] I. Altintas, A. Memon, B. Ludäscher, "Design and Execution of Scientific Workflows using Web Services", Web Services Meeting, San Diego Software Industry Council (SDSIC), Jan. 2004.

[5] SEEK: Science Environment for Ecological Knowledge, http://seek.ecoinformatics.org

[6] SPA: http://kepler.ecoinformatics.org/spa.html

[7] GEON: Cyberinfrastructure for the Geosciences, http://www.geongrid.org