Solution Service Composition for Analysis of Online Science Data

Sara Graves, Rahul Ramachandran, Ken Keiser, and Manil Maskey
University of Alabama in Huntsville
301 Sparkman Drive
Huntsville, AL 35899

Christopher Lynnes and Long Pham
Goddard Earth Sciences Data and Information Services Center
Goddard Space Flight Center
Greenbelt, Maryland

Abstract - The Information Technology and Systems Center (ITSC) at the University of Alabama in Huntsville has teamed with the Goddard Earth Sciences Data and Information Services Center (GES DISC) to develop and demonstrate the use of distributed web service technologies for science data analysis solutions. Many science data repositories are now providing online access to vast amounts of data. This project is demonstrating an approach of composing workflows of distributed web services to provide reusable analysis solutions for these online data resources. The GES DISC is providing an operational data environment for the resulting demonstrations.

I. Introduction

The NASA ACCESS project, titled "Deployable Suite of Data Mining Web Services for Online Science Data Repositories" is addressing the development of technology that will allow users to locally define analysis workflows that can be executed on data residing in online repositories. The existing ADaM Toolkit is a mature package of data analysis components that have typically been utilized by researchers for data available locally. This project, called Mining Web Services (MWS) as a short name, is enabling ADaM capabilities for use in a distributed web service environment. Further the project has investigated ways to compose these services into solution services that chain together existing services to solve a particular problem and is then reusable by other programs that need the same solution. This approach aims to extend the use of complex analysis techniques to researchers who may not have expertise with some of the lower level analysis steps.

II. Background

The MWS project was structured to take advantage of several existing resources that provided capabilities that are important to the effort. These included web services technology that provides a standard way to remotely interface with programmatic components and to orchestrate the chaining of services through standardized service descriptions. The ADaM Toolkit has been developed and expanded over several years and provides a rich set of analysis tools. The Earth Science Markup Language (ESML) [1, 2] provides the ability to programmatically interpret heterogeneous data formats, thus support data interoperability. The GES DISC has a large online repository of publicly available scientific data sets as well as existing internal process automation capabilities.

A. ADaM Toolkit

The ADaM (Algorithm Development and Mining) toolkit [3, 4, 5, 6] was originally developed in 1994 with the goal of mining large scientific data sets for geophysical phenomena detection and feature extraction, and has continued to be expanded and improved. ADaM provides knowledge discovery and data mining capabilities for data values, as well as for metadata. Thus, unlike most data mining software, ADaM has been designed for use with scientific and image data from the outset. ADaM includes not only traditional data mining capabilities such as pattern recognition, but also image processing and optimization capabilities, and many supporting *data preparation* algorithms that are useful in the mining process. Recently, ADaM was redesigned as a toolkit of discrete, independent components for the evolving computing landscape which can be used together in different combinations to perform many complex tasks. This redesign also allows the algorithms in ADaM to be easily packaged as grid or web services [7] and is being extensively used by different research groups.

B. GES DISC (Goddard Earth Sciences Data and Information Services Center)

GES DISC provides users with the ability to mine data at the archive where the data are stored [8]. This avoids the need to deliver large volumes of data to the user for large-scale mining activities. The Near Archive Data Mining (NADM) system was first implemented for the Tropical Rainfall Measuring Mission. It allows users to run data mining or other data reduction algorithms of their own design either on newly acquired data, or by pulling data out of the archive in a data mining campaign. This system was later implemented for data from the Terra and Aqua satellites. This data mining capability has been successful at enabling several data mining and reduction activities that would not otherwise be possible. However, the requirement for the user to supply the algorithm has limited its use to a select few “high-end” users. It has long been a goal of the GES DISC to make data mining algorithms available to a broader community. In fact, *Lynnes and Mack, 2001* proposed a scenario by which third-party data service providers (such as UAH/ITSC) would enable users to run algorithms at data archives. MWS is striving to provide this capability in a standardized, cost-effective way.

III. Approach

For some time, the University of Alabama in Huntsville (UAH) Information Technology and Systems Center (ITSC) has been investigating the use of distributed services for use with data mining, subsetting, image processing, thematic map generation and other spatially oriented data applications. The mining web services architecture (see Figure 1) builds on this expertise to provide a deployable web service suite, hardened for widespread use and deployable to data provider sites for execution of user scenarios against large data stores.

A. Web Services

A web service is a software component designed to be used between distributed machines following standard internet protocols. Web services are the basic components of an SOA.

The two main styles of web services are Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). SOAP provides a standard message protocol for communication based on XML. SOAP web services have two main conventions: any non-binary attachment messages must be carried by SOAP and the service must be described using Web Service Description Language (WSDL). SOAP was selected primarily for this WSDL feature, which has particularly useful for driving the composition of complex workflows. SOAP has become so popular that the terms “SOAP” and “web service” are often used interchangeably.

1. Mining Web Services

ADaM components are implemented using standard C++ for portability across platforms. These discrete components can already be scripted together in a variety of ways to solve complex problems. This project will augment this existing technology by repackaging ADaM components as SOAP-based web services for easier and more dynamic integration in emerging distributed service oriented architectures. The suite of mining and related services will be packaged into deployable bundles to allow easy deployment in the web server environments at online data provider sites. An incremental development approach will be utilized, with different ADaM algorithms selected for three major staggered releases based on their utility and the requirements of the target demonstration deployments. The “deploy early and deploy often” philosophy will be used in releasing these packages to prospective users with an additional set of algorithms added at every new release. Also, a registration mechanism for advertising the different service capabilities of a package via ECHO (Earth Observing System Clearinghouse) [9] will be created.

B. SOA

A Service Oriented Architecture (SOA) packages together independent services that have well defined interfaces. These interfaces are designed such that they are interchangeably invoked by applications or other services implemented on heterogeneous platforms and languages. Some benefits of developing applications within an SOA include re-use of existing software, reduced deployment time, minimal code changes reduces chance of introducing errors, and orchestration of data and computational resources at distributed locations.

C. Service Workflows (chaining)

A variety of approaches for orchestrating solution service workflows were explored, but MWS concentrated on using BPEL (Business Process Execution Language). The use of BPEL was chosen primarily because it has been well received by many communities and applications so it approaches the level of a standard, and as a result of this acceptance there are a number of evolving tools available for this technology. Acceptance of a common execution language allows for the portability of workflow definitions across more systems and hopefully easier acceptance at other data centers as the benefits are demonstrated.

As part of this project, the suite of mining and related web services will be deployed at existing NASA data provider sites which will demonstrate the utility of this package in a variety of environments. The utility of this system to users will be the ability to define and execute service workflows, comprised in part of the services deployed at NASA data centers. The proximity of the services, such as data mining, image process, etc., at the data centers will allow access to large amounts of NASA data for user-defined analyses. These analyses workflows themselves can then be exposed as solution services for other researchers to utilize, assuming permission by the originators, resulting in potential for a rich set of functionality that researchers can deploy against large amounts of data without the overhead of local storage or processing resources.

D. Implementation

The initial implementation of web services for existing ADaM tools has been completed by ITSC. The implementation included placing a SOAP wrapper around each ADaM operation. ITSC considered many options of the SOAP implementation before deciding on SOAP::Lite [10].

SOAP::Lite is the Perl implementation of SOAP. Although Java's implementation of SOAP, Axis [11], provides a wide variety of tools that assist in implementing services and clients, Java was not appropriate in this case because the ADaM tool was implemented in C++. ITSC had encountered numerous issues while interfacing Java Native Interface (JNI) to C++ for other ITSC projects. The Axis C++ offers a C++ implementation of SOAP, however, it has not been well-accepted by the software industry so far. The Python implementation of SOAP, pySOAP, has not been kept up-to-date. Therefore, due to ITSC's experience on other projects and the familiarity of Perl at Goddard Space Flight Center (GSFC), ITSC decided on SOAP::Lite for Perl as the web service implementation language.

Once UAH finished implementing the web services, the Web Service Definition Language (WSDL) for each of the services were published. Then each of the web services was tested for interoperability using Perl and Java clients.

In order to allow the data center full control over enforcing local policies for file creation, the SOAP wrappers specify the full path of the input file only, plus additional parameters as necessary. The data center constructs the output directory and filename and returns it to the calling program in the SOAP response. This may then be used as the input file to the next step in the workflow. This interface mechanism allows the data center to control where files are created on its system and to implement safeguards against inappropriate usage.

1. Solution Service Composition

A set of operations is often necessary to provide an appropriate solution to complex science problems. When using SOA to solve science problems, solutions to such complex problems can be designed using workflows that employ various web services. A workflow describes how tasks are orchestrated, what components performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are being tracked.

Currently, the industry standard for service orchestration is the Business Process Execution Language (BPEL). BPEL provides a standard XML schema for workflow composition of web services that are based on SOAP. There are other workflow composition tools that create workflow descriptions for a set of web services execution, however, the tools are not standardized yet. This standardized composition description is eventually deployed on a BPEL engines. A BPEL engine is required to process the instructions described in BPEL. Most of the BPEL engines are open source software, such as the ActiveBPEL engine [12]. Various visual workflow composers are also freely available. The problem with having different BPEL engines is that even though the BPEL is a standard language, these engines require an additional deployment descriptor that describes engine specific additional dependencies. Thus, the deployment descriptor varies from engine to engine. Here, deploying a BPEL translates to taking a BPEL description and asking the engine to expose the workflow as a web service. Being able to expose a composition of a set of web services makes the solution itself a web service and thus reusable. It is important to note that the BPEL engine can be totally isolated from the services being utilized.

Ideally, for science problems involving large data, data centers should be capable of hosting web services so that the data can be accessed easily, keeping the data sent over the network to minimum. The only required component at a data center to host web services functionality is an appropriate web server. Dropping the set of web services to the web server container should make the web services available for use.

ITSC investigated various freely available workflow composers and their associated engines. The two BPEL engines that stand out are: ActiveBPEL and Sun's BPEL engine. ActiveBPEL Designer and NetBeans IDE (v5.5) provide the composer for the corresponding BPEL engines respectively. These composers also provide mechanism to create deployment descriptors and user friendly deployment interfaces. ITSC also investigated the BPEL Execution Engine (BEXEE). However, BEXEE is no longer being supported. FiveSight PXE BPEL Engine [12] has also been investigated.

During the investigation, it was discovered that NetBeans IDE was the best in terms of the usability in creation of workflows. However, ActiveBPEL, provided an easier mechanism to deploy the workflows. For BEXEE and PXE, there were no specified workflow composers. However, deployment descriptors were manually created for the BEXEE and PXE engines. Manual creation of deployment descriptors was error prone. We found that the BPEL descriptions were transportable between all of these engines. As the result of our investigation, we chose ActiveBPEL as our BPEL engine. Furthermore, a plan was made to develop a mining-specific application for an easier and more user friendly interface to the creation of BPEL and its deployment descriptor. This application alleviates the necessity for users to have extensive knowledge of BPEL.

2. Graphical Workflow Composition

The Active BPEL Designer requires too much in-depth knowledge of BPEL definitions to be useful for mining users.

To assist users in composing the workflows, ITSC has adapted a graphical composition tool to work in this environment. The Xbaya composition tool was developed by a team at Indiana University, led by Dr. Dennis Gannon [13], in conjunction with the NSF-funded LEAD project (Linked Environments for Atmospheric Discovery). ITSC is also a participant in the LEAD project.

ITSC extended the Xbaya software to adapt it for use with composing mining workflows. These changes include:

- Ability to deploy workflows to a user provided generic BPEL engine
- Ability to save workflows as BPEL descriptions
- Ability to save the workflow as a simple batch file
- Ability to invoke recently deployed BPEL workflows

An example workflow presentation from the MWS Workflow Composer is shown in Figure 2.

3. Sandbox Environment

Since GES DISC's Near Archive Data Mining (NADM) is an operational data center, it is not the ideal site for the highly iterative process of developing a data mining "solution". Instead, ITSC is developing an external sandbox environment that allows the user to experiment with a small subset of the data, ADaM algorithms and associated parameter settings. Once the user is satisfied with a solution definition, it can be invoked on large quantities of data within GES DISC's NADM system. The user will be able to mine interactively, or to request a mining subscription, which will run on newly produced or acquired data when it is ingested at the GES DISC. The interfaces and security mechanisms between the "sandbox" and data provider defined and implemented for this deployment can be used in other similar situations, requiring the integration of ADaM services into an existing processing system.

4. Demonstration Architecture

The architecture is shown in Figure 3. A workflow is generated within the Workflow Composer, based on experimentation in a sand box. This workflow is then deployed to a BPEL engine (Flow #1), which returns a URL pointing to the WSDL for that workflow (Flow #2). This URL is then transmitted to the GES DISC via a web services request along with a specification of the data to be mined, such as the dataset to be mined and temporal or spatial constraints (3). This request is provided to the GES DISC's processing engine (4), also known as the Simple, Scalable Script-based Science Processor for Measurements (S4PM) [14] for data mining at the data archive. The S4PM engine is responsible for acquiring the data from the archive, then executing the requested workflow on each input file. However, note that the workflow itself is not provided to S4PM. Instead, S4PM uses the supplied WSDL URL to fetch the WSDL document and then invoke the corresponding web service in the BPEL engine (5), supplying the full path of the input file. The BPEL engine turns around to invoke the atomic web services in the proper order at the GES DISC (6). Finally, the output is transmitted to the end user from the data center (7).

5. Demonstration

ITSC and GES DISC have worked together to produce a demonstration web service workflow involving the orchestration of mining web services at a client site, using services and data available at the repository.

The demonstration includes a solution to a real science problem of creating a cloud mask in Geostationary Operational Environmental Satellite (GOES) images. The solution involves the use of K-means clustering algorithm to label each data point as either part of a cloud or not part of a cloud. There were preprocessing services to convert the generalized satellite format (GSF) cloud data to Attribute-Relation File Format (ARFF) format, which is the allowed data format for K-means operations. Furthermore, the result was post-processed using another web service to convert into an image file for visualization.

Mining Workflow Composer was used to define the workflow. The ActiveBPEL engine is hosted on a different computer. This demonstration began with the construction of the workflow in the sand box (Figure 2) using a three-step process:

1. Importing the WSDL for required web services using the Mining Workflow Composer.

2. Creating the workflow with the imported services using the modified Xbaya tool as the Mining Workflow Composer.

3. Deploying the workflow to the BPEL Engine using the Mining Workflow Composer.

The result of this workflow, a URL to the WSDL, was transmitted along with the dataset, start time, and stop time via a SOAP request to the Data Mining Service at the GES DISC. The Data Mining Service deposited the request in the S4PM engine, which proceeded to acquire the data files from a local archive. As each file was acquired, it was processed according to the workflow by invoking the corresponding web service presented by the BPEL engine. The results were staged to an FTP directory, from which the user ITSC retrieved them and verified the output.

6. Current Status

An initial prototype addressed the scientific problem (cloud mask) described in the demonstration section above. The

GES DISC has successfully hosted the mining web services developed at ITSC by deploying them at the GSFC's web service container. WSDLs for the web services hosted at the GSFC have been made available. ITSC has successfully tested those services using simple clients generated using the WSDLs. Externally generated workflow descriptions have been created and successfully executed through service interactions with the GES DISC.

A second prototype developed around a science scenario for the detection of dust storms is in progress.

IV. Conclusions/Results

Initial prototyping efforts have been successful and encouraging for the use of web service workflows in providing a mechanism where remote users will be able to deploy custom data analysis solutions at data repositories. This approach will allow users to run these analyses on large amounts of data available at the repositories, while still maintaining control on the details of what services are used and the parameters employed. The solution workflows themselves can then be exposed as web services for other researches to further incorporate in other studies.

Additional investigation is underway to prototype how effective this approach can be for more complicated data analysis workflows.

V. Acknowledgments

We gratefully acknowledge funding from NASA's ACCESS program for this project. In addition, ADaM was originally developed under a NASA Research Announcement, and later refactored as part of a NASA AISRP grant.

VI. References

[1] R. Ramachandran, S. A. Christopher, S. Movva, X. Li, H. T. Conover, K. R. Keiser, S. J. Graves, and R. T. McNider, "Earth Science Markup Language: A Solution to Address Data Format Heterogeneity Problems in Atmospheric Sciences," Bulletin of the American Meteorological Society, accepted 2004a.

[2] R. Ramachandran, S. Graves, H. Conover, and K. Moe, "Earth Science Markup Language (ESML): a solution for scientific data-application interoperability problem," Computers & Geosciences, vol. 30, pp. 117-124, 2004b.

[3] T. Hinke, J. Rushing, S. Kansal, S. J. Graves, H. S. Ranganath, and E. Criswell, "Eureka Phenomena Discovery and Phenomena Mining System," presented at AMS 13th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography and Hydrology, 1997a.

[4] T. Hinke, J. Rushing, H. S. Ranganath, and S. J. Graves, "Target-Independent Mining for Scientific Data: Capturing Transients and Trends for Phenomena Mining," presented at Proceedings Third International Conference on Data Mining (KDD-97), Newport Beach, California, 1997b.

[5] T. Hinke, J. Rushing, H. S. Ranganath, and S. J. Graves, "Techniques and Experience in Mining Remotely Sensed Satellite Data," Artificial Intelligence Review: Issues on the Application of Data Mining, vol. 14, pp. 503-531, 2000. [12] ActiveBPEL, The Open Source BPEL Engine, http://www.activebpel.org/, Jun, 2006.

[6] R. Ramachandran, H. Conover, S. J. Graves, and K. Keiser, "Algorithm Development and Mining (ADaM) System for Earth Science Applications," presented at Second Conference on Artificial Intelligence, 80th AMS Annual Meeting, Long Beach, Long Beach, CA, 2000.

[7] J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch, and A. Lin, "ADaM: A Data Mining Toolkit for Scientists and Engineers," Computers & Geosciences, vol. 31, pp. 607-618, 2005.

[8] C. Lynnes and R. Mack, "KDD Services at the Goddard Earth Sciences Distributed Active Archive Center," in Data Mining for Scientific and Engineering Applications, R. L. e. a. Grossman, Ed.: Kluwer Academic Publishers, 2001, pp. 165-181.

[9] ECHO, NASA's Earth Observing System Clearinghouse: http:// www.echo.nasa.gov , accessed October 2006.

[10] SOAP, SOAP::Lite for Perl: http://www.soaplite.com/ , accessed October 2006.

[11] AXIS, Apache AXIS SOAP Implementation: http://ws.apache.org/axis/ , accessed October 2006.

[12] ActiveBPEL, "The Open Source BPEL Engine", http://www.activebpel.org, June 2006.

[13] Y. L. Simmhan, S. L. Pallickara, N. N Vijayakumar, and B. Plale, "Data Management in Dynamic Environment-driven Computational Science", IFIP Working Conference on Grid-Based Problem Solving Environments (WoCo9), 2006.

[14] C. S. Lynnes, "The Simple, Scalable, Script-based Science Processor", (in press), Earth Science Satellite Remote Sensing, Springer-Verlag, 2006.

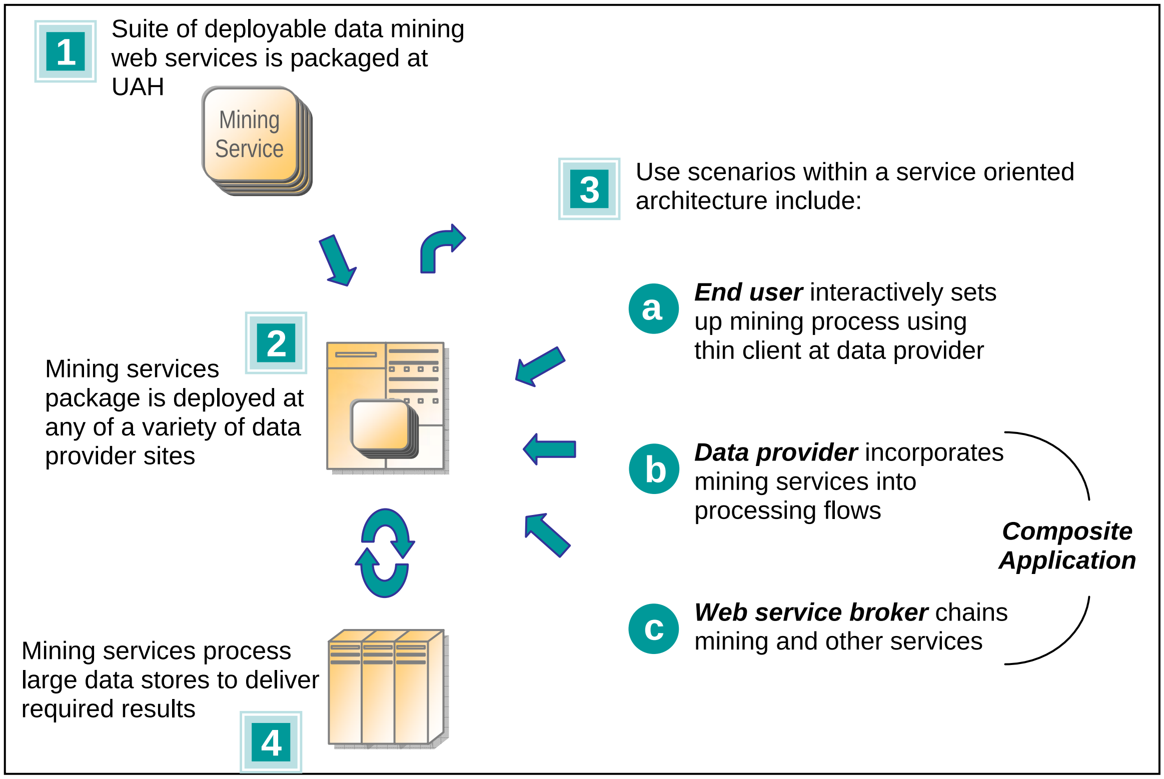

Figure 1: Overall Architecture of Mining Web Services

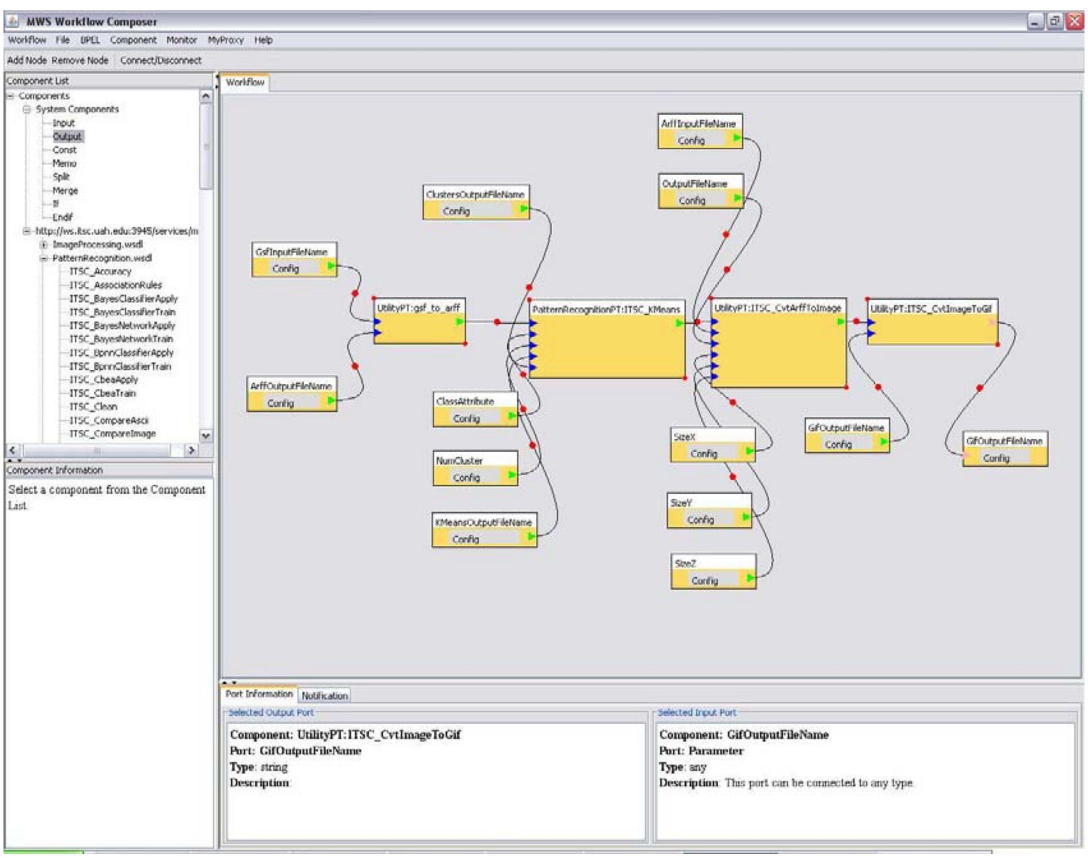

Figure 2: Mining Workflow Composer based on Xbaya

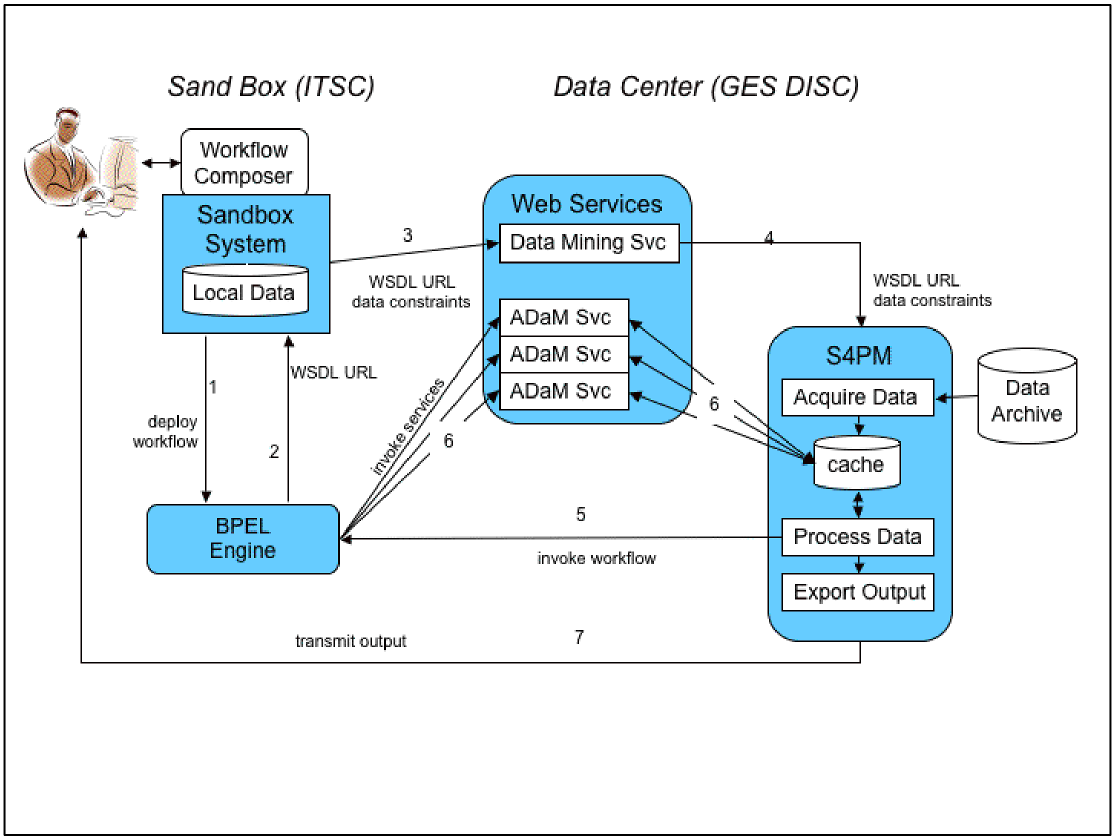

Figure 3: Demonstration Architecture