

BPEL Workflows Combining Standard OGC Web Services and Grid-enabled OGC Web Services

Tino Fleuren, Paul Müller
University of Kaiserslautern
PO Box 3049
D-67653 Kaiserslautern, Germany
{fleuren, pmueller}@informatik.uni-kl.de

Abstract

When designing a Grid workflow, it might be necessary to integrate different kinds of services. In an ideal scenario all services are Grid-enabled. But real workflows often consist of Grid-enabled and non Grid-enabled services.

One reason is that Grid-enabling services can be costly. Therefore it is favorable to solely Grid-enable the compute-intensive and time-consuming applications. Additionally, workflows should be allowed to include Grid jobs that execute legacy applications.

Another reason is that very often, third parties charge fees for accessing their services. Hence, it is impossible to convert such a third party service into a service that can be integrated into a Grid environment at all.

This paper discusses problems of designing a workflow that consists of all these different kinds of services. The geospatial domain is exemplarily used to demonstrate difficulties that workflow designer have to overcome, i.e. constructing a geospatial workflow by using combinations of conventional Web services (XML-based), standard OGC Web services and Grid-enabled OGC Web services (WSRF-based).

The concept of a workflow engine capable of enacting these workflows is presented and an implementation based on the ActiveBPEL engine is proposed.

1. Introduction

From the designer's perspective, designing a geospatial workflow should be an easy task. The construction should be supported by graphical tools and without knowledge of the different kinds of technologies like conventional Web services (XML-based), standard OGC Web services (OWS) or Grid services (WSRF-based).

These services can be contemplated as small units of

functionality that implement parts of geospatial applications. The flexible composition of those building blocks forms large new applications that comply with the requirements in typical geospatial scenarios.

Some years ago, the middleware performing all these user needs would have to be implemented using conventional programming languages. To overcome these obstacles Web service orchestration languages were defined, that describe workflows which can be executed on a workflow enactment engine.

The Business Process Execution Language (BPEL) can be used to orchestrate Web services based on XML. BPEL offers the means to build workflows with SOAP-based Web services. However, it is neither supposed to invoke Grid services nor OGC Web services.

The Open Geospatial Consortium (OGC) has specified open interfaces and protocols to make spatial information and services accessible in a standardized way. The Geographical Information System (GIS) community broadly adopted these specifications. Some of the applications based on the OGC specifications are highly compute-intensive, time-consuming and need powerful computation resources that a Grid environment can provide.

In order to Grid-enable the geospatial Web services, OWS have to be adapted to suffice the conditions of the Web Services Resource Framework (WSRF), which is not, at present, supported by OGC. Because of the existing libraries that implement numerous of OGC Web services, the expense of converting all these functionalities would be enormous. The GIS community has misgivings about the cost and amount of work that arouse. Moreover, it would be an advantage to be able to integrate legacy applications as a Grid job in a BPEL process without converting them at all.

Other services offer functionalities that are not time critical or that do not need the security settings of a Grid environment. For example, test operations examine the conditions whether a time-consuming computation would deliver

the desired result. In several scenarios, these operations do not need to be executed on the powerful compute nodes of a Grid and therefore do not have to be converted to Grid services.

Some organizations provide data like maps or offer external, fee-required services that the user wants to integrate into his workflow, too. Generally, these services cannot be used in a Grid environment via standard Grid protocols.

Hence, a workflow that merely consists of Grid services would not suffice to fulfill the typical scenarios. The user should be free to use OGC Web services and Grid services that comply with the Web Service Resource Framework (WSRF) simultaneously.

After giving a short overview on the various technologies, a categorization of the different kinds of services that come into play in a geospatial workflow are categorized and the problems of porting geospatial Web services in Grid environments are discussed. At the end, this paper presents a concept of a holistic workflow enactment engine and an implementation based on the ActiveBPEL engine by ActiveEndpoints installed in a Globus Toolkit 4 environment of the SDI (Spatial Data Infrastructure)-Grid project¹.

2 Related Work

Several aspects of the corresponding problems have been examined. [13] addresses the problems of using standard OWS in a workflow and of Grid-enabling OWS. [18] integrates Grid services in BPEL processes by enhancing the interfaces of the services, which cannot be done for every service, e.g. third party services. [16] shows an example BPEL process that invokes a Grid service but which is complex and cannot be designed without knowledge of the different technologies that are involved. [8] presents a BPEL enhancement with a new BPEL activity that can invoke Grid services. The authors have developed an enhanced workflow enactment engine that complies with the Grid security constructs.

All these research projects concentrate on different aspects but do not consider a workflow engine that integrates both Grid services and non-Grid services.

3. Technologies

This section briefly introduces the different technologies that are involved in typical geospatial workflow scenarios.

3.1. Geospatial Services

Geographic Information Systems (GIS) are used to solve geographic problems and to represent geographically ref-

¹This work is funded by the German Federal Ministry of Education and Research (BMBF) (D-Grid Initiative).

erenced information. The geospatial community collects, manages and uses geographic data.

In order to access GIS in a standardized way, the OGC specified several types of geospatial Web services.

The client posts their requests to an OGS Web service using HTTP. The encoding for the OGC Web service operations can either be XML (HTTP POST/SOAP) or keyword-value pairs (HTTP GET/remote procedure call). All OGC Web services are able to describe themselves via the *GetCapabilities* operation. The response to a *GetCapabilities* request is an XML description.

The Web Feature Services (WFS) provide access to stored geographical features, i.e. a user defined geographic piece of interest that can be modeled or represented using geographic data sets. Examples of geographic features include streets, sewer lines, accidents, etc. In addition to the *getCapabilities* operations, a WFS provides the *describeFeatureType* function which retrieves the XML schema to allow the client to parse the result sets and the *getFeature* function which actually performs the query of the feature data. The WFS returns a Geography Markup Language (GML) document, which is an XML grammar to express geographical features.

The Web Coverage Services (WCS) handle raster geospatial data. The coverages are objects within a geographical area. A WCS can be used for discovery, query or data transformation operations. Examples of different data formats supported by a WCS are: DTED, GeoTIFF, or NITF.

The Web Map Service (WMS) are used to display and integrate various layers of geographic datasets onto the same map. A WMS provides a standardized access to maps rendered in a format such as PNG, GIF or JPEG by using the operations *getMap* and *getFeatureInfo*. Most of the GIS manufacturers support the WMS interfaces. The optional WMS *getFeatureInfo* operation returns a more detailed description of objects included in the maps.

The Web Processing Services (WPS) offer GIS calculations covering processing and analytical functions. A WPS can describe any geospatial function including all of its input and output parameters and activate its execution. The WPS define the *describeProcess* operation that returns a description of a process including inputs and outputs and the *execute* operation that performs the calculations and returns the result.

WMS, WFS, WCS are not described by WSDL documents. WPS does support SOAP and WSDL, but this is not yet exploited by the GIS community. Therefore, the present OWS cannot be used in BPEL workflows.

3.2. Grid computing

The vision of Grid computing is to provide a system where different computational resources and data storages can be linked together via standard protocols in a common infrastructure. According to Ian Foster, the Grid “coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service”[12].

With the introduction of the service-oriented computing paradigm and the Web service standards like WSDL and SOAP, and with their broad acceptance in the industry, the Open Grid Forum (OGF) developed the Open Grid Services Architecture (OGSA) to define a standardized, open architecture for Grid-based applications.

While OGSA proposes a higher-level architectural perspective on service-oriented Grid computing, the Web Services Resource Framework (WSRF) is a description of the environment required to implement the OGSA model. WSRF is a set of specifications that define the modeling and management of stateful resources using existing Web services technologies. WSRF defines the interfaces of Grid services which address issues like dynamic service creation, lifetime management, notification, and manageability that enable applications to interact with Grid services in standardized ways.

One key contribution of Grid computing is to enable and simplify collaboration among a wider audience. The users of the Grid can be grouped dynamically into a number of virtual organizations (VO). These VOs can combine their resources to build a larger Grid. The privacy of the offered resources must be guaranteed, so that only authorized users can access them. Hence, security is one of the major requirements for a Grid. The Grid can enforce security rules among the VOs and implement policies. The Grid security model has to enable the interoperability between administrative domains, the creation and management of VOs and must integrate with existing security architectures, models and implementations.

3.2.1 Globus Toolkit and Security

The Globus Toolkit 4 (GT4) is a software system that can be used for building Grids [1]. Version 4.0 complies with the WSRF specifications. The Grid Security Infrastructure (GSI) is the very part of GT4 that provides security functionalities. GSI is based on public key cryptography and uses extensions of X.509 certificates to identify persistent entities such as users and hosts.

GSI uses a third party, a Certificate Authority (CA), to guarantee the ownership of each of the public keys. The owner must secure the corresponding private keys and never reveal them to the public. GSI offers four aspects of secu-

rity:

Transport-Level and Message-Level Protection Transport Level Security (TLS) and Message Level Security (WS-Security and WS-SecureConversation) are used for signing or encrypting messages.

Authentication X.509 certificates are used to prove the identity of a user.

Authorization The authorization framework allows for several authorization constructs, e.g. a “grid-mapfile” access control list, an access control list managed by a service, a custom authorization handler and access to a authorization service via the Security Assertion Markup Language (SAML) which provides a means to describe and to exchange security data.

3.2.2 Delegation

In order to directly interact with a remote service, the client uses their certificate to prove their identity. However, in a Grid it is possible to distribute jobs to remote Grid machines and let them distribute their child jobs to other machines under the client’s security policy by using delegation. Delegation is implemented by creating a new, temporary public/private key pair, building a new certificate containing the public key and signing it with the client’s long-term private key. These temporary certificates are called proxy certificates. Delegation reduces the number of times the user has to reenter their credentials.

When a job is submitted, the proxy certificate, the private key for the proxy and the user certificate are sent to the Grid service. When the job proves its identity to another service it provides the proxy certificate and the standard certificate. Then it can use the chain of certificates to prove that it is entitled to use the user’s subject name (SN).

A proxy certificate is a compromise in security terms because it is possible for a thief to impersonate another user by using the private key as long as the certificate is valid which is usually within a time interval of a few hours.

3.2.3 Job execution

Globus Toolkit 4 contains a Web service-based Grid Resource Allocation and Management (WS GRAM) component. WS GRAM is a WSRF-based Web service used by computational resources to remotely submit, monitor, and cancel jobs.

A WS GRAM job is defined by an XML document (see figure 1) and can consist of a single-job description to start a single execution of an application or a multi-job description used to trigger several jobs simultaneously. WS GRAM

provides secure job submission to many types of job scheduler for users who have the right to access a job hosting resource in a Grid environment. WS GRAM offers a uniform and flexible interface to batch scheduling systems, such as PBS, Condor, LSF, and SGE.

Additionally, staging activities can be described to occur before or after the job, i.e. it can be specified where input and output data can be retrieved or copied. If staging is requested in the create call, suitable delegated credentials must be passed. The staging credential gives WS GRAM the right to interact with the data management services that copy input data to the resource node.

```
<job>
<executable>/bin/bash</executable>
<argument>/geo/geospatial-script.sh</argument>
<stdout>${GLOBUS_USER_HOME}/geo.stdout</stdout>
<fileStageIn>
<transfer>
<sourceUrl>gsiftp://gridnode.de/s.dat<sourceUrl>
<destinationUrl>file:///tmp/s.dat</destinationUrl>
</transfer>
</fileStageIn>
</job>
```

Figure 1. Example of a simple job description file.

3.3. Service Orchestration

With the introduction of the Web service technologies — WSDL (Web Service Description Language) to describe the interfaces of the services; SOAP to exchange messages between them, independently of the underlying protocol; and, UDDI (Universal Description, Discovery, and Integration) to publish and discover them — applications based on Service-Oriented Architectures (SOA) can be quickly composed [11].

The description of the interaction between Web services at message level, including the control flow and the data flow, is called service orchestration [14]. Business Process Execution Language for Web Services (BPEL or BPEL4WS) [5], Web Service Choreography Interface (WSCI), and Business Process Management Language (BPML) are examples of Web service orchestration languages or specifications.

In this paper the focus lies on BPEL because it is a de-facto industry standard (approved by the OASIS consortium) that is supported by multiple vendors (e.g. Microsoft, IBM, Sun Microsystems, etc.). Additionally, there are several open source engines available like ActiveBPEL by ActiveEndpoints.

BPEL can be used to construct complex workflows. The language is based on XML and supports flow control, asynchronous communication and business process transactions.

The language produces applications that are modular, scalable and reusable by using standards like Xpath, XSLT and XQuery to manipulate the XML documents. A BPEL process is made of three main entities:

- The partners that represent the services involved in the workflow.
- The variables used to manipulate the data (SOAP messages) exchanged between partners and to save states of the process.
- The activities that describe the control flow, such as invoking a Web service, assigning a value to a variable, or executing several activities in sequence or parallel.

The workflow will be instantiated by a workflow enactment engine. This instance is itself a Web service and can be included in other workflows. WSDL documents describe the interfaces of the workflows and the integrated services.

The next section discusses the problems aroused by integrating these different technologies.

4. Integrating different types of services in BPEL processes

For building the most flexible workflows, a service-oriented geospatial system should not restrict the granularity of (Web) services that can be integrated. So, one possibility to categorize the services is their granularity. The grain size can range from coarse (for example another workflow) to fine (for example a component that must be combined with others to create a complete geospatial workflow).

Another possibility to categorize the services relates to the technology that they are based on. The new system envisioned in this paper (see figure 2) will support the following types of services:

- The standard Web services described by WSDL documents. BPEL was designed to orchestrate these services, so they can be easily integrated.
- The WSRF-based Grid services are, in contrast to the standard Web services, stateful. They are addressed by dynamic endpoint references (EPR) which are not supported by BPEL.
- The WS GRAM services are WSRF-based Grid services that allow to submit and monitor jobs. By using these WS GRAM services, some of the legacy geospatial applications can then be used as executables in a Grid job.
- The OGC Web services do not (yet) support the WSDL specification. Therefore, they cannot be integrated into a BPEL process.

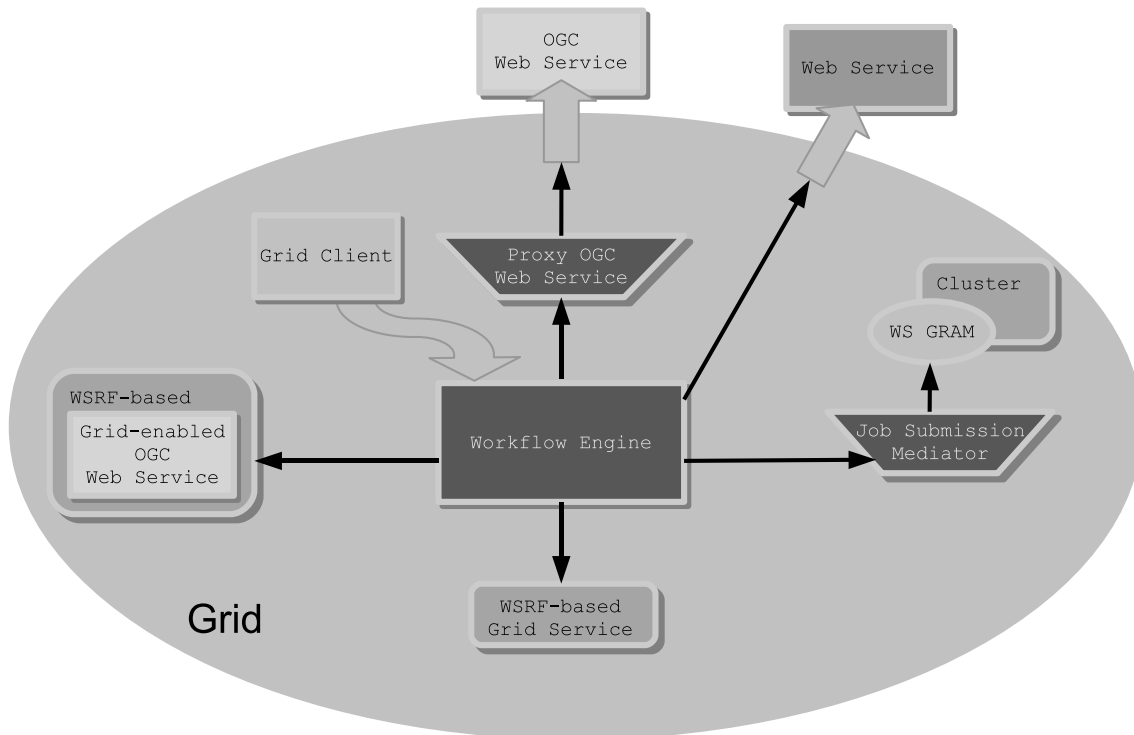


Figure 2. A workflow engine integrating different types of services

In this section possible ways of how the envisioned system can integrate these different types of services is discussed.

One aspect concerning the integration of OGC Web services in a Grid environment is the question of which geospatial services need to be Grid-enabled. There are three conceivable categories:

- **Compute-intensive OGC Web services:**
These services require powerful computation resources that a Grid environment can provide and therefore should be converted to Grid services.
- **OGC Web services for data dissemination:**
These services are often provided by third parties and cannot be accessed by standard Grid protocols. Therefore, they cannot be integrated in the Grid environment.
- **Supportive OGC Web services:**
These services implement supportive task that are quick and do not need the computational power of a Grid. At the moment, a great many implementations are not WSDL-based.

Figure 2 presents the envisioned workflow engine that is capable of integrating all these different types of services.

The next section discusses the difficulties of integrating the diverse technologies.

4.1. BPEL and OGC Web services

SOAP is the message exchange format for WSRF-based Grid services. OGC Web services use several formats like XML (WFS), binary (WMS and WCS) and ASCII text (WCS). The OGC is currently debating support for SOAP for its Web service specifications.

Hence, in order to integrate an OGC Web service into a BPEL process, it is necessary to establish communication between BPEL and geospatial services.

[13] propose a solution that introduces an intermediary SOAP node which acts as a proxy (see figure 2: proxy OGC Web service) between the geospatial Web service and other SOAP nodes. Using such a proxy Web service provides a way to integrate OGC Web services into the BPEL process. All requests to an OGC Web service will be intercepted by the proxy Web service and translated to the appropriate OGC format.

The raw binary format poses another problem when receiving the result of a WCS *getCoverage* request. This can be solved in two ways:

The first option is to embed the raster dataset within a SOAP message as binary. The ActiveBPEL engine is capa-

ble of including binary data in the SOAP message by base64 encoding the binary data in a CDATA section element. Another method is to add the binary data as a SOAP attachment [4]. In geospatial workflows and other e-science domains huge amounts of data are to be dealt with. So it is not desirable to copy large amounts of binary data to the BPEL engine, because the data is often the input of the next process activity and has to be copied to the next activity.

The second possibility is to enhance the proxy Web service in a way that it stores the binary data and returns only a handle (like an URL) to the data [13]. The next BPEL activity, which can be another proxy Web service, uses the handle to retrieve the data. The proxy Web service can be configured to either delete the intermediate data or to keep it persistent, so that it could be reused if need be.

One possible database that would be perfectly suitable to store the intermediate result is PostGIS [2]. PostGIS is an open source geographic information system software that enhances support for geographic objects to the PostgreSQL object-relational database. OGSA-DAI¹ can be used to access and integrate PostGIS in the Grid environment. Figure 3 shows the example invocation of a WCS within a BPEL process. First the BPEL invokes the proxy WCS which translates the request and calls the `getCoverage` operation of the real WCS. The WCS returns its result to the proxy WCS which then stores the data in the PostGIS database. After that it returns the handle to the data to the calling BPEL process.

A separate proxy Web service will have to be developed for each type of OGC Web service. These proxy Web services will then be integrated in an enhancement of the ActiveBPEL engine.

4.2. BPEL and WSRF-based Web service

In contrast to Web services, which are stateless, the WSRF-based Grid services are stateful. The combination of a Web service and a stateful resource is called a WS-Resource [7].

The WSRF specification defines a WS-Resource to be a Web service that clients can use to access the state of a resource and manage its lifetime.

WS-Addressing specifies a construct called an endpoint reference (EPR) which can be used to address a Web service endpoint [17]. Besides the URI of the Web service, an endpoint reference can contain more information.

It is very common to use a factory pattern to instantiate a WS-Resource. A factory is a Web service offering an operation to create resources and assign a resource key to this resource. The resource key is a unique ID to identify the resource in later uses and associates the resource with a Web service. Hence, the client invoking the stateful resource has

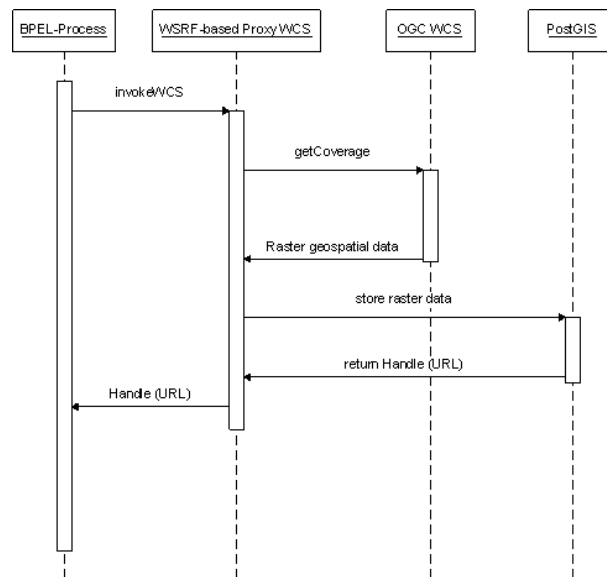


Figure 3. Integrating a WCS in a BPEL process.

to know the URI of the associated Web service and the resource key to reach the desired resource, which is known as a WS-Resource-qualified endpoint reference [15].

BPEL is not designed to invoke stateful Web services and consequently there is no standardized way to handle the returned resource key.

Several solutions to this problem have been discussed. Three approaches are:

4.2.1 Passing the endpoint reference as an operation parameter

It is possible to shift the problem from the workflow description to the Grid services by enhancing the Grid services in a way that the workflow engine can pass the endpoint reference received by the factory to the Grid service as an operation parameter [16].

In this solution the interfaces of all existing Grid services that are to be used in the workflow will have to be enhanced. However, it is often not possible to do so, especially when integrating third party Grid services. So the proposed workflow engine will not implement this solution.

4.2.2 Using standard BPEL activities to extract endpoint references

Another solution to this problem is to design a BPEL process that updates WS-Addressing fields to address the dynamic endpoints [18].

¹OGSA-DAI: <http://www.globus.org/toolkit/docs/4.0/techpreview/ogsadai/>

When designing a BPEL process that invokes a Grid service, the first activity is to start the creation of the resource by invoking the *createResource* operation of the resource factory service. A stateful instance of the resource with an ID (the resource key) is created and associated to a Web service as outlined in the implied resource pattern. The next step is to assign the values of an input variable and to set the WS-Addressing fields with the unique identifier information. After that, BPEL invokes an operation on a resource-aware Web service by using the resource key.

In a typical workflow that scientists might wish to execute, several hundreds of services may be included [10]. Even if a scientist wishes to include one Grid service, the knowledge of the internal constructs of WSRF-based Grid services and technologies like SOAP are implied which results in workflows, that only specialists can design. Therefore, this approach is too difficult and time-consuming. The proposed workflow engine will not implement this solution.

4.2.3 Grid-specific BPEL extensions

In [9] a BPEL extension is described that allows the interaction with both stateless and stateful services. The authors added the following BPEL activities:

- *gridCreateResourceInvoke* and *gridDestroyResourceInvoke* are used to create and destroy a WS-Resource.
- *gridInvoke* is used to call a Grid service.

First a resource is instantiated by the resource factory service. *gridCreateResourceInvoke* internally calls the factory service that instantiates the WS-Resource and returns the resource key which is stored internally. The *gridInvoke* uses the internally stored resource key to call the Grid service.

[9] enhanced the ActiveBPEL engine, so that it can execute the new activities and additionally developed a graphical BPEL designer tool based on an Eclipse plugin. They implemented a SOAP handler that checks if the SOAP messages contain resource key information.

Their strategy is analogous to that described in section 4.2.2 but transparent to the BPEL process designer.

In a later version of their BPEL enhancements, the workflow engine is capable of operating in security-related settings, especially the GSI of the Globus Toolkit, i.e. transport-level and message-level protection, authentication and authorization, and delegation. Their solution consists of adding some security information in the message header of the client's SOAP call. The user's existing proxy certificate is transferred with the SOAP header to the workflow engine. The proxy certificate can either be pre-generated or retrieved from a MyProxy Credential Management Service [6].

A major advantage of using the MyProxy repository is the feature to automatically renew the proxy certificate if it expires while the job is still running, which is important when integrating long-running Grid services.

The proposed workflow engine will be based on the BPEL extensions of [9].

4.3. BPEL and WS GRAM

The Globus Toolkit 4 offers the possibility to execute jobs on Grid nodes, which provides the possibility to use legacy geospatial applications, i.e. command line based applications, in the Grid environment. The job description determines how the data and the application are transferred to the appropriate Grid node, where the application is executed.

The WS GRAM services are WSRF-compliant. Therefore, the workflow engine needs to be WSRF-enabled (see section 4.2).

The data of a WS GRAM job is published as part of a WSRF resource. A job factory is called in order to create job instances (as a WSRF resource).

The Managed Job Factory Service (MJFS) creates the jobs and interact with the particular resource manager. The created job instance is called Managed Executable Job Resource (MEJR). Additionally, there is an analogous service/resource implementation for multi-jobs.

The envisioned workflow engine must be capable of passing a job description to the MJFS and to subscribe for job notifications.

This paper proposes to manage the Job submission by using a new Grid service: the Job Submission Mediator (JSM) (see figure 2). The JSM receives the job description and calls the MJFS or forwards the job to the particular job scheduler (see figure 4). Then it listens to the job notifications and translates them. When the completion status is signaled, the result is transferred to the workflow engine.

In addition, the JSM is responsible for monitoring the job execution and reacting by sending an appropriate status message or error message to the BPEL process.

4.4. Grid-enabling OGC Web services

In order to integrate existing algorithms for geospatial calculations into a Grid, it is necessary to implement a new WSRF-based Grid service that imports and integrates the libraries containing the algorithms. Therefore, the existing code libraries implementing geospatial algorithms can be reused, but have to be ported into the new environment.

It is obvious that writing a new Grid service requires expenses like time and manpower. Hence, it is often necessary to solely Grid-enable those algorithms that really need the kind of performance that a Grid provides.

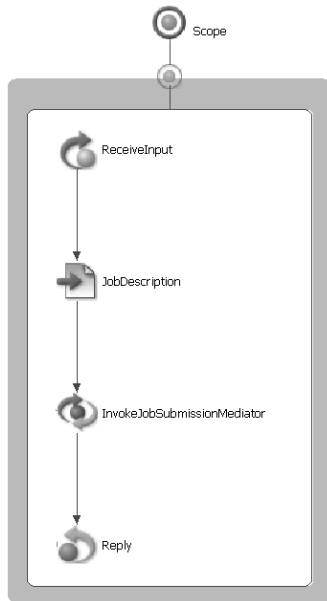


Figure 4. Example BPEL process executing a WS GRAM job

The Globus Toolkit offers the possibility to implement its Grid services using the C, Python, and the java programming language. These options ease the integration of existing libraries.

5. Summary and Future Work

The proposed geospatial workflow engine will be based on the ActiveBPEL engine, and will be able to:

- integrate OWS by using proxy Web services,
- store intermediate data in a PostGIS database,
- use the BPEL enhancements of [9] which can include secure WSRF-based Grid services and the ability to use delegation and renewal of proxy certificates,
- submit a job to a WS GRAM (and such the underlying cluster) by using a job submission mediator which manages the communication to the WS GRAM and which is capable of handling notifications.

Several other features can be imagined. In some workflows the integration of human interaction could become necessary. So it would be interesting to use BPEL4People enhancements [3].

After successfully designing and executing a workflow, the user would like to save the workflow description and to

publish it to his peers, so they can execute the same workflow with their data. Therefore, reusability is another issue that a geospatial workflow environment should be able to offer.

References

- [1] Globus Toolkit (GT4). <http://www.globus.org>.
- [2] PostGIS. <http://postgis.refractor.net/>.
- [3] WS-BPEL Extension for People (BPEL4People), 2007. <http://www.active-endpoints.com/documents/documents/1/BPEL4People-v1.pdf>.
- [4] ActiveEndPoints. ActiveBPEL Business Process Execution Engine. <http://www.active-endpoints.org>.
- [5] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. S. and S. Thatte, and I. Trickovic. Business Process Execution Language for Web Services version 1.1. Technical report, BEA, IBM, Microsoft, SAP, Siebel Systems, 2003.
- [6] J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository: Research Articles. *Softw. Pract. Exper.*, 35(9):801–816, 2005.
- [7] K. Czajkowski, D. F. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe. The WS-Resource Framework, June 2005. <http://www.globus.org/wsrp/specs/ws-wsrf.pdf>.
- [8] T. Dörnemann, T. Friese, S. Herdt, E. Juhnke, and B. Freisleben. Grid Workflow Modelling Using Grid-Specific BPEL Extensions. In *Proceedings of German e-Science Conference 2007*, pages 1–9, 2007.
- [9] T. Dörnemann, M. Smith, and B. Freisleben. Composition and Execution of Secure Workflows in WSRF-Grids. In *8th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 08)*, page (accepted for publication). IEEE Press, 2008.
- [10] W. Emmerich, B. Butchart, L. Chen, B. Wassermann, and S. L. Price. Grid Service Orchestration Using the Business Process Execution Language (BPEL). *J. Grid Comput.*, 3(3-4):283–304, 2005.
- [11] T. Erl. *Service-oriented Architecture*. Prentice Hall, 2005.
- [12] I. Foster. What is the Grid? A Three Point Checklist, 2002.
- [13] P. J. Gobe Hobona, David Fairbairn. Workflow Enactment of Grid-Enabled Geospatial Web Services. In *Proceedings of the 2007 UK e-Science All Hands Meeting*, 2007.
- [14] C. Peltz. Web Services Orchestration: A Review Of Emerging Technologies, Tools, and Standards. Technical report, Hewlett Packard, 2003.
- [15] B. Sotomayor and L. Childers. *Globus Toolkit 4: Programming Java Services*. Morgan Kaufmann, December 2005.
- [16] K. L. L. Tan and K. J. Turner. Orchestrating Grid Services using BPEL and Globus Toolkit 4. In *Proc. 7th PGNet Symposium*, pages 31–36, 2006.
- [17] W3C. Web Services Addressing (WS-Addressing), 2004. <http://www.w3.org/Submission/ws-addressing/>.
- [18] M. Zager. SOA/Web Services - Business Process Orchestration with BPEL, 2005. <http://webservices.syscon.com/read/155631.htm>.