A SOA-based environment supporting collaborative experiments in e-Science

Andrea Bosin, Nicoletta Dessì, Bairappan Madusudhanan, Barbara Pes
Dipartimento di Matematica e Informatica,
Università degli Studi di Cagliari,
Via Ospedale 72, 09124 Cagliari, Italy

Pre-print submitted to International Journal of Web Portals (IGI Global)

ABSTRACT

There are many sophisticated environments that allow creating and managing scientific workflows, whereas the workflow itself is provided as a service. Scientific Grids handle large amounts of data and deal with sharing resources, but the implementation of service-based applications that use scientific infrastructures still remains a challenging task, due to the heterogeneity of Grid middleware and different programming models. This paper proposes to support scientists with an e-Science environment providing functionality in a simplified way, especially for communities whose IT skills are not so smart not only to consider the Grid a source of computational power, but also an information infrastructure. To promote both integration among components and user interaction and leverage existing work in both business and scientific environments, the paper outlines a SOA-based scientific environment where a scientific experiment is modeled through an abstract workflow defining the functional model of the experiment. In turn, the workflow tasks are mapped to the corresponding scientific services by a workflow engine, the key being to separate logical aspects from implementation issues. Services depend on the type of experiment and can be re-used, or wrapped, or moved straight into a new workflow. Infrastructural services discover suitable resources that match user requirements and schedule workflow tasks to the selected resources. Further, they monitor the execution of each single task and aggregate the results of the execution. The proposed approach provides a simple-to-use and standardized way for the deployment of scientific workflows in a distributed scientific environment, including the Grid.

Keywords: E-Science, Grid, Workflows, Service Oriented Architectures, Web Services

INTRODUCTION

Collaboration in scientific experiments, obtained by sharing data, tools, and expertise towards a common scientific goal, is becoming more and more appealing for e-Science, thanks to the availability of Information and Communication Technology (ICT) methods and tools. In particular, the Service Oriented Architecture (SOA) paradigm is attractive since it can effectively support distributed cooperation. However, the heterogeneity and dynamicity of services and of their underlying infrastructures make the aspects of creating valuable complex service environments an emerging research issue in the scientific community.

Advances in computing technologies have enabled scientists to validate new research practices in many scientific fields and to evolve from individual activities to work conducted in teams, exploring research issues at time and space scales both greater and finer than ever before. This new research context is becoming more and more complex in terms of the number of collaborating researchers, the diversity of computing environments supporting collaborative efforts among each participant in data/computation intensive applications, the number of emerging powerful and effective data analysis tools enabled by new technologies, the distribution of data and computing resources and the consequent orchestration of the data analysis tools across various platforms.

Indeed, the computing resources available to a scientific experiment, the network capacity, connectivity and costs may all change over time and space since some components are added, removed or temporary unavailable. Similarly, the scientist may move from one location to another, joining and leaving groups of researchers and frequently interacting with computers in changing experimental situations. In short, the research environment we consider is constantly in evolution and scientific collaboration keeps on increasing the aggregation and sharing of heterogeneous and geographically dispersed resources. In practice, this means that computation does not occur at a single location in a single context, but rather spans a multitude of situations and locations covering a significant number of heterogeneous hardware or software components.

E-Science is the term usually applied to the use of advanced computing technologies to support scientists. In short we can say "e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it" (De Roure, 2004). The above definition is still to come at the structural level: technical problems limit the usability of the e-infrastructure presently in production, i.e. the Grid, whose technology is still far from allowing a true interoperability of scientific applications and/or computational experiments. As a consequence, the level of detail needed for the successful deployment of scientific applications on the Grid still remains very high. Moreover, scientists want to get work done and they do not want to deal with the complexity of building applications that expose details of the underlying e-infrastructure. They must be able to express their problem by composing application specific components in an easy-to-use, easy-to-re-use and easy-to-modify form. Their favorite model of programming is to compose a workflow by means of a graphical interface via "drag-and-drop", and they loathe writing "programs" in XML. However, the visual programming model must be sufficiently powerful to address a wide range of conditions, exceptions, iteration and adaptive control.

The paper aims at defining the needs and the building blocks for the next step in the advance of e-Science environments. Grids and distributed systems, augmented with various management capabilities, are considered essential aspects of the e-Science environment. To promote both integration among components and user interaction, the paper proposes to extend the use of solutions developed for business environments and in particular the adoption of a Service Oriented Architecture. An architectural model for the deployment of scientific workflows is presented as well as a case study to validate the effectiveness of the proposed approach.

The paper is structured as follows. Section II reviews some related works. Section III presents an overview of the infrastructures supporting scientific collaboration. Section IV gives a short overview of the scientific workflows requirements. Section V presents the proposed architectural approach while Section VI gives some implementation details for the execution of BPEL-based scientific workflows on heterogeneous platforms, including the Grid. Section VII shows a case

study in the field of data mining in which Web Services are combined to carry out a data mining process. Finally, conclusions are drawn in Section VIII.

RELATED WORK

E-Science workflow tools have been built to address a wide spectrum of applications, ranging from basic tools that are designed to handle tasks such as simple data analysis and visualization to complex workflow systems that are designed to run large-scale e-Science applications on remote Grid resources. These systems need to support multiple concurrent users, deal with security requirements, and run workflows that may require the use of a sophisticated layer of services (Fox, 2006). For example, my Experiment (Goble, 2007) is an open repository for items arising in scientific workflows and experiment plans. That repository has been established collecting a significant set of scientific workflows, spanning multiple disciplines and multiple workflow systems, built according to Web 2.0 design principles. As another environment, my Grid (*http://www.mygrid.org.uk*) is a suite of tools designed to "help e-Scientists get on with science and get on with scientists". The tools support the creation of e-Laboratories and have been used in diverse domains such as systems biology, social science, music, astronomy, multimedia and chemistry. The tools and the infrastructure allow the design, editing and execution of workflows in Taverna (*http://www.taverna.org.uk)*, the sharing of workflows and related data by myExperiment (*http://www.mygrid.org.uk/tools/myexperiment*), the cataloguing and annotation of services in BioCatalogue (*http://www.mygrid.org.uk/tools/biocatalogue*), the creation of user-friendly clients such as UTOPIA (*http://utopia.cs.manchester.ac.uk*). These tools help to form the basis for the team's work on e-Labs. (*http://www.mygrid.org.uk/tools/e-labs*).

McPhillips (2009) identifies desiderata for scientific workflow systems – namely clarity, predictability, report ability, and reusability. Moreover, ease of composition and editing, the ability to automatically log and record workflow enactments and the flexibility to incorporate new tools are all-important features (Fox, 2006). The interoperability aspects of scientific workflow systems are addressed in Elmroth (2010) that investigates differences in the execution environments for local workflows and those executing on remote Grid resources. A complete overview of features and capabilities of scientific workflow systems is presented in Deelman (2009).

There are a number of widely recognized Grid workflow projects. Many of these began life in the "desktop" workflow space, but they have evolved over time to address the large-scale e-Science applications. A Grid-aware framework for the construction of distributed workflows and their management and execution is provided by systems like Triana (Taylor, 2005), Kepler (Pennington, 2007), Pegasus (Deelman, 2005), and ASKALON (Fahringer, 2007). Specifically designed for the life sciences, Taverna (Oinn, 2007) has been the first system to recognize the importance of data provenance and semantic Grid issues. Based on BPEL (*http://www.oasis-open.org/committees/wsbpel*), QoWL (Brandic, 2006) and GPEL (Slominski, 2007) are significant examples of workflow systems designed for dynamic, adaptive large-scale e-Science applications.

In particular, Deelman (2009) recognizes BPEL as the de facto standard for Web-Service-based workflows. The use of BPEL for Grid service orchestration is proposed as foundation in Leymann (2006) since it fulfils many requirements of the WSRF standard (*http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf*). The appropriateness of BPEL is also examined and confirmed in Chao (2004), Dörnemann (2007) and Emmerich (2006). These works mainly focus on scientific workflows and rely on extending or adapting BPEL, thus creating

dialects. While developed for the business domain, technologies like BPEL are then recognized suitable to address the requirements of e-Science applications in Akram (2006), supporting the composition of large computational and data analysis tasks that must execute on remote supercomputing resources. Bosin (2010) presents an architectural model for the deployment of scientific workflows using BPEL, while Bosin (2011) discusses the challenges encountered in the seamless integration of BPEL processes within an e-Science infrastructure.

INFRASTRUCTURES SUPPORTING SCIENTIFIC COLLABORATION

Collaboration is essential for combining approaches, combining skills, and sharing resources and the concept of scientific experiment is rapidly moving from the idea of a local laboratory activity towards a computer-based process involving complex data analysis.

Workflow systems provide specialized computing environments for automating this process allowing scientists to represent experimental stages without the hassle of focusing on computational resource management. Formally, a workflow is a computer program composed by a set of tasks that the researcher orchestrates according to her/his experimental methodology without being aware of the complexity associated with managing and deploying applications.

E-Science workflow systems have been built to address a wide spectrum of applications, ranging from basic tools that are designed to handle desktop tasks such as simple data analysis and visualization to complex workflow systems that are designed to run large-scale e-Science applications on heterogeneous and distributed resources including the Grid. Like in the past, a typical experimental scenario requires data to undergo several processing stages, launching the computations and storing the output results, but a workflow system makes it much easier to automate the process of accessing and using distributed resources.

There is therefore a need for providing functionality in a simplified way, especially for scientific communities whose IT skills are not so smart not only to consider the Grid a source of computational power, but also an information infrastructure. Towards these needs, Service-Oriented Computing (SOC) is an emerging paradigm that may open a completely new way for e-Science applications. In this paradigm, applications are built by assembling together independent computational units, called services. A service is a stand-alone component distributed over a network, and made available through standard interaction mechanisms. An important aspect is that services are open, in that they are built with little or no knowledge about their operating environment, their clients, and further services therein invoked. This aspect enables researchers to vision a scientific workflow as composed by granular services allowing large-scale collaboration, easy access to very large data collections and distributed computing resources.

As services will become easily available, researchers can bring them together, without being concerned about the applications or products involved in delivering the service. This ability of selecting and assembling together heterogeneous services, namely the service orchestration, heavily depends on which information about a service is made public, on how to choose those services that match the user's requirements, and on their actual run-time behavior. Virtualized in the form of services, software applications may be accessed using well-defined high-level interfaces while many scientific infrastructures support low-level interfaces to computing resources, often limited to simple batch job submissions.

Currently, there are many sophisticated environments allowing creating and managing scientific workflows, whereas the workflow itself is provided as a service. However, building service-based applications that use scientific infrastructure still remains a challenging task, due to the heterogeneity of Grid middleware and different programming models. In recent years,

distributed systems and Grid technology integrated many computing resources. This makes it possible to carry on experiments where very high performance computing ability and large-scale dataset are required. Scientists of today routinely rely on computers and information sharing over the Internet to aid them in their research. Often, scientific progress necessitates large-scale international collaboration; examples such as the human genome project and particle physics experiments would not be feasible without it. The term e-Science refers to this type of large-scale cooperation. However, the Grid didn't realize the full promise of being the best computing infrastructure for e-Science and it is not adopted by a large category of scientists who prefer to choose and harness collaborative tools and technology (that often provide less efficient solutions than Grid applications) and rely on them to design their experiments. Grid computing is better suited for scientific organizations with large amounts of data being requested by a small number of users (or few but large allocation requests); on the other hand there is a large number of researchers, namely naïve researchers, requesting small amounts of data (or many but small allocation requests).

We distinguish two classes of Grids. The first class consists of General Purpose Grids (GPG) that provide computing and data resources to a broad class of application communities: EGEE, TeraGrid, Open Science Grid, etc. The second class of Grids are those devoted to a specific Scientific Domain (SDG) or technical application field, such as bioinformatics, geosciences, chemical informatics, earthquake science, astronomy, etc. In the first category, the use of services is based on providing the basic elements of security, data management, and remote job execution and information services. In the second category we find more specialized services including application services, user-level metadata services, data discovery services and specialized workflow tools.

According to what we experienced, GPG users are not able to identify and, consequently, define computational challenges wide enough to (saturate) the amount of computational resources made available. They prefer to control the successful execution of their own applications and seem unable to change the way research is done by adopting, for instance, cooperative and interdisciplinary approach. A major problem is related to the communication difficulties among different scientific communities that are requested to change their research practices in terms of the vision that each community has, what it is offering and what it wants to receive from the other scientific communities. On the other side, when there is some agreement in different research communities, GPG seem not deliver the promise of better applications and usage scenarios because of the complexity associated with managing and deploying applications.

The next step for supporting e-Scientists is to provide them with an e-Science environment (in addition to the infrastructure) that comprises high level services, which may be easily and directly accessible while hiding the infrastructure that is changing at run-time. Towards these needs, Service Oriented Architecture (SOA) may open a completely new way for e-Science applications by enabling the researchers to vision the entire research process as composed by granular services allowing large-scale collaboration, easy access to very large data collections, the use of computing resources, etc.

SCIENTIFIC WORKFLOW FEATURES

In this section, we introduce the fundamentals on scientific workflows that are relevant to our work. Usually, scientists compose, launch and monitor their workflows, each of which consists of a set of tasks that produce and/or consume data. Being each task a specialized data processing

activity, dependencies among tasks are created by the need for data to be produced before it is consumed.

Since tasks that can be accessed through the network, a natural way to improve their accessibility is to turn them into services providing uniform access to computational resources, tools and automated service discovery. Services correspond to different functionalities that encapsulate technical capabilities such as:

- Authenticate and authorize use of resources
- Submit, monitor and control tasks inside workflows
- Move a data set to and from the computing resource, including to and from the desktop
- Publish a data set, specifying global name and attributes
- Locate a data set by global name or by data set attributes
- Account resource usage
- Monitor and control the aggregate system (system administration, user views)
- Advanced reservation of resources

Such functionalities can be easily encapsulated into web services that have their counterparts in the e-commerce or business-to-business (B2B) world where one must discover resources, query capabilities, request services, and have some means of authenticating users for granting access to resources and accounting their usage. In many ways, the requirements for service-based e-Science environments do not differ substantially from those of business environments. Then, focusing on the service architecture required to support e-Science, the question is: do we need to provide entirely new solutions or can we adopt (reuse) solutions developed for B2B environments?

In this paper we explore the latter option, and borrow many SOA concepts and standards from the business domain. A first benefit of this approach is almost evident: the SOA framework and in particular web services are based on widely accepted standards and supported by many software tools, both open source and commercial. However, in the case of e-Science, there are a number of issues that are significant departures from the classic B2B scenario. The primary difference stems from the fact that enterprise workflows are about repetitive business processes and science is based on experiments. While experimentation has a significant repetitive component, the scientist is constantly altering the pattern of a workflow because that is where discoveries are made. A second difference arises from the fact that scientific users require workflows fitting a variety of domains. This forces scientific workflows to be composed of heterogeneous tasks, each dealing with different requirements (i.e. fast database access, high-performance resources, computer graphic facilities, data streaming, etc…). Related to these differences the following specific issues must be addressed.

Modular Structure and Composability - Each scientific workflow is associated to and operates on relevant information that may consist of a combination of tasks and data collected from several resources. The traditional scientific user spends a substantial amount of time managing remote data files and resources. Web services are deployed to manage data and replicas of data automatically and all data products, including those that are intermediate results, can be automatically saved to be reused in a related workflow or to restart a workflow that had a flaw in a downstream component. As well, sub-workflows can be saved for later reuse.

Monitoring - Workflow tasks are related and linked together. In some sense, the workflow is a whole of single collaborative procedures that express the experiment strategy. Often, the scientific workflows that run on distributed Grid resources result in long-running processes and having data services that can retain the intermediate results generated by each workflow step is essential. It is also essential to have mechanisms to track a failed step in a workflow, suspend the action and make a call to the resource broker to allocate new resources and then restart the workflow.

Context Sharing and Reuse - One of the foundations of e-Science is the requirement that experiments are repeatable and that all derived data products are traceable back to their sources. This suggests that the tasks making up the process should be annotated, thus the experiment can be understood, repeated and shared easily. Metadata may be considered to describe each data product, each single task and to capture all context information, including input and output data. Authored metadata documents can be stored in a repository to be automatically indexed for efficient retrieval. This provides the opportunity of reusing some tasks and repeating the whole experiment by other users who are "skilled in the art".

Classification - Workflows must be described by proper classification and placement with respect to the collaborative scientific community, thus the experiments can be identified, classified and browsed by the research community members.

Moreover, scientific workflows use and collect lots of data that are distributed on heterogeneous computing environments. To make Grid technologies more widely usable, we devise the need for promoting wider integration with computational infrastructures (such as clusters, desktops, P2P networks etc…) that can be more easily shared.

Scientific services: the new paradigm

E-Business organizations and e-Science environments have many elements in common, but the question is if and how existing business models can support distributed scientific experiments. Several approaches have been proposed for collaborative scientific environments, but an extensive analysis in devising a mechanism for designing and implementing scientific experiments in a collaborative environment is still missing.

The range of accessible technologies useful to support scientific experiments can be classified broadly into these categories:

1. Toolkits specifically aimed at supporting experiments, with friendly and usable interfaces;
2. Software tools that are not specifically designed to support experiments, but that are still essential in enabling them (e.g., mathematical computation tools, data mining tools, data warehousing tools);
3. Methods to ensure data privacy;
4. Widely deployed infrastructures that may be useful in scientific experiments, such as Web services and Grid computing.

Hence, the problem is the integrated use of heterogeneous applications and software tools that were not designed specifically to promote interaction and cooperation, but still are inherently suitable for cooperation support. This scenario is similar to that of enterprise environments, whose progress requires large-scale collaboration and efficient access to very large data collections and computing resources. Although sustainable interoperability models are emerging

for market players (such as service providers, stakeholders, policy makers, and market regulators), they are currently deployed mostly in areas where high computing power and storage capabilities, usually needed by scientific environments, are not mission-critical. Applying emerging web service technology to the scientific environments takes a flexible and multi-faceted approach: it aims at assessing task-user-system functionality incrementally according to the continuous evolution of scientific cooperative environment.

The challenge is to define services supporting a scientific environment whose basic characteristics are as follows:

1. *Efficiency*: services enabling discovery and provisioning of resources free the scientist from low-level technical and repetitive work and it contributes to the creation of "best practices" eventually valuable, comparable and shared with other people.
2. *Reproducibility*: In scientific computations, service execution occurs multiple times on the same or different instances of data by users belonging to external organizations.
3. *Re-use and automatic enhancement of knowledge*: services produce outputs that form new (potential) inputs for other scientific processes triggering a virtuous re-cycle mechanism that incrementally increases knowledge.
4. *Traceability*: in executing a scientific experiment, data sources can be traced and checked a priori.

OUTLINING A SERVICE-BASED SCIENTIFIC ENVIRONMENT

The definition of a scientific workflow is typically entrusted to a human actor (but it is possible to think to entrust this task, at least partially, to an expert system) who has the competence of application domain (we will call this actor workflow designer and she/he will be the figure of the researcher) and that selects the distributed resources and their composition, without necessarily attending the implementation of the low level technical aspects.

Workflow systems are designed to run large-scale e-Science applications on distributed heterogeneous resources. As such, they need to support multiple concurrent users, deal with security, and run workflows that may take days to months to complete.

In this section, we outline a SOA-based architectural solution aiming at offering a communication bridge between the heterogeneous computational environments used to develop and host scientific applications. The idea is to outline a SOA-based scientific environment for implementation and deployment of pluggable "experiment handlers" supported by web services. The following basic directions feature the above outline.

Integration of scientific applications - A large category of scientific applications tend to be self-contained, isolated pieces of software for which interoperability is not an issue. The user spends a substantially amount of time in managing data integration since this is often done manually. The advantage of a SOA approach is that scientific applications can be expressed in terms of web services. In our research environment, for example, they are simply plain Java applications with a web service interface - implemented through JAX-WS (*https://jax-ws.dev.java.net*); in the more general SOA approach the transformation can be achieved through suitable adapters and/or wrappers (Papazoglou, 2007). Most important, web service technology has been designed to promote seamless integration and interoperation of services.

Access and usability of resources - Computing facilities available to scientific applications address a large spectrum of resources. At one end of the spectrum are desktop/laptop hardware and software for simple data analysis and visualization. At the other end we find resources

organized in clusters managed by some kind of lower level scheduler, e.g. LSF (*http://www.platform.com/workload-management/high-performance-computing*) or SGE (*http://www.sun.com/software/sge*), working for the Grid middleware configured on top of it, i.e. gLite (*http://glite.web.cern.ch/glite*) or Globus (*http://www.globus.org*). Needless to say, the user interfaces of such schedulers are completely different, authentication is based on different types of user credentials (e.g. user/password or X509 certificates) and the user is often required to log to a remote system to perform job submission. If the access to resources is abstracted by means of a web service interface, the user interaction results almost completely decoupled from the low level details (hardware, operating systems, middleware, schedulers). Web service wrappers over the existing software assets (i.e. schedulers or middleware) are developed if needed.

Workflow languages and engines - Many incompatible languages and engines are available for workflow design and enactment and the choice is strongly influenced by the model adopted for problem at hand, since the workflow has to interface both with applications and resources. In our model, the choice of the language is a natural consequence, since SOA has its own standard, namely BPEL. Originally designed for service orchestration in business domain, BPEL is the de facto standard for Web-Services-based workflows and gained much attention from scientific communities.

SOA are inherently multi-tier architectures (Papazoglou, 2007), and clients (users or applications) typically interact with the public abstract upper layers. Upper layers are built on the facilities offered by lower layers, which are usually private and hidden to the clients. In the proposed architecture the process layer is the topmost and is populated by all the BPEL workflows relating to the e-Science domain. Its clients access a BPEL workflow instance as a standard web service, but internally it relies on the facilities offered by finer-grained web services, mainly business services and infrastructure services.

Business services (BS) represent scientific applications or parts of application and are implemented by application developers in such a way to interact both with users (business interface) and infrastructure (infrastructure interface) only through a web service interface. The implementation is as much as possible decoupled from the details of the resources on which it will be running. Role and responsibilities of BS include accepting user invocation both according to the request-response pattern and one-way request pattern with asynchronous notification, notifying its availability, performing infrastructure activities such as monitoring or clean up.

Infrastructure or bearing services provide all the necessary facilities for resource allocation, access and management, data management, etc. through their web service interface (infrastructure interface).

IMPLEMENTATION ASPECTS

In this section we briefly validate the feasibility of the proposed service-based environment. As previously mentioned, a scientific experiment is modeled through an abstract workflow defining the functional model of the experiment. The workflow tasks are mapped to the corresponding scientific service by the workflow engine, the key being to separate logical aspects from implementation issues. Services depend on the type of experiment and can be re-used, or wrapped, or moved straight into a new workflow. Analogously, it is possible to include in the workflow services belonging to external organizations, hence achieving collaboration, knowledge sharing, and externalization of procedures.

If web service and BPEL standards are adopted, writing the suitable XML documents can perform workflow design, deployment and enactment. Since this is not feasible for the average

researcher, the availability of a graphical tool is a strong requirement. Among those freely available in Internet, we have chosen Netbeans IDE 6.5.1 (*http://netbeans.org*); for completeness, we can cite a few others like Eclipse (*http://www.eclipse.org*) or JDeveloper (*http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html*).

Bearing services discover suitable resources that match the user requirements and schedule workflow tasks to the selected resources. Further, they monitor the execution of each single task in the resource and aggregates results of the execution. There are invoked by a user i.e. by a human or by other applications such as BPEL workflows. To enable the use of Grid and other distributed resources through different access protocols, we implemented the following services:

- The *Resource Allocation* (RA) service implements the user interface for resource management. It accepts user requests for resource allocation/release, notifies users when resources are available, keeps track of resource endpoint and status, and coordinates other infrastructure services. Allocation is done either directly or, depending on the resource type, delegating to the corresponding RM (see below).
- The *Resource Manager* (RM) service is a wrapper around the user interface of legacy schedulers. Its role is to accept requests from the RA, verify user credentials, and dispatch them to the underlying scheduler. The Security Manager (SM), a part of RM in our implementation, performs user authentication. The SM, in turn, can query a legacy Authentication Manager (AM) such as a LDAP server. We have implemented RMs both for LSF, SGE and gLite.
- The *Business Service Proxy* (BSP) is a SOAP intermediary for routing user request messages. Its role is to accept all user requests directed to business services and route them to the service endpoint (which may be on a private network), if necessary it can route the service response back to the user.
- The *Notification Proxy* (NP) is another SOAP intermediary. It is responsible for routing the notification messages generated by the business services to the user.

To reduce time and effort needed to interface our web services with Grid middleware, we developed bearing services on top of jLite (*http://code.google.com/p/jlite*), a Java library providing a high-level Java API with functionality similar to gLite shell commands while hiding complexity of underlying middleware and its configuration. The library is pure Java and can be used on any Java-capable platform. Current implementation supports complete gLite job management lifecycle including VOMS proxy creation and delegation, transfer of job input files, job submission, job status monitoring and retrieval of job output files. Normal, collection and parametric gLite jobs are supported.

Among the business services, we have implemented a data mining service based on the Weka library (Hall, 2009) as a Java JAR application whose only requirement is Sun JRE (*http://www.oracle.com/technetwork/java/javase/downloads/index.html*). Such a service will run virtually on any resource (both the application and the JRE are downloaded and installed on the fly at run time) and is considered in the next section.

Fig. 1 shows the validated service-based environment with the main connections between services and the other components.

Fig. 2 instead presents the UML sequence diagram describing the interactions between a BPEL instance, the bearing and business services, and other legacy (non-web) services.

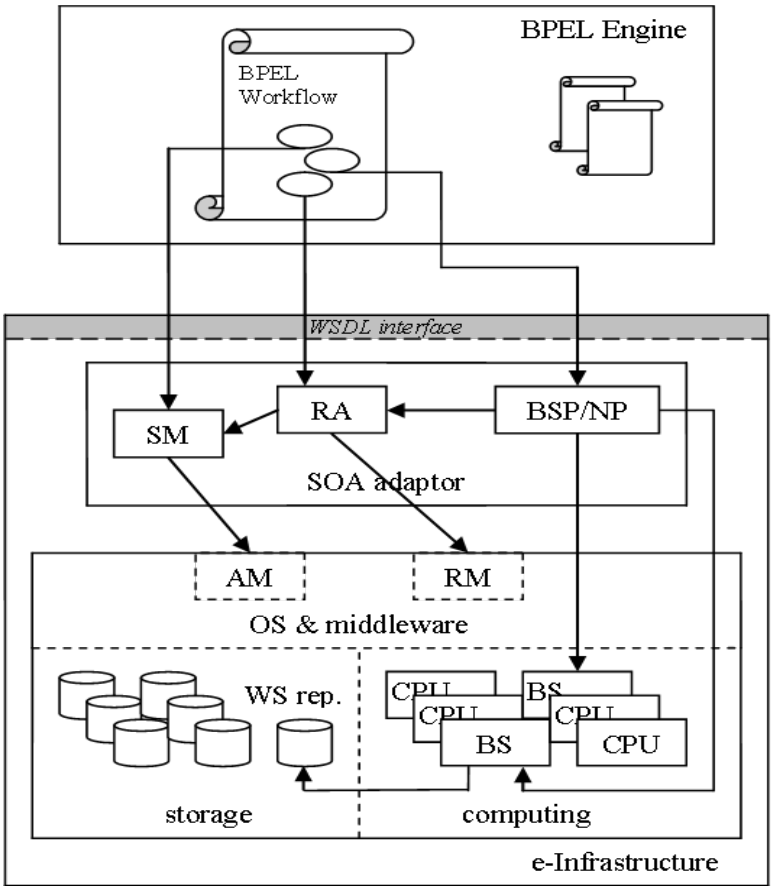

Figure 1. The validated service-based environment.

EXECUTING EXPERIMENTS

As a sample usage scenario we consider the following data mining experiment described by a BPEL workflow and executed by a BPEL engine (running on the user desktop or somewhere else). Given training and test set (i.e. a matrix whose rows represent samples and columns are features) compare the accuracy of two different classifiers for an increasing number of selected attributes (by some attribute selection algorithm). The resulting accuracy is visualized in a bar chart diagram on the user desktop and is updated during the computation as the number of attributes increases. Both the data mining and the visualization application can be provided by the user or downloaded for our public repository as standalone JAR web services, which can be executed using Sun JRE.

The resources required to perform the experiment are the following (and are specified by the user in the workflow input file):

1. the user desktop running the visualization service (the service endpoint is provided), statically allocated
2. one resource for every instance of the data mining service (the URL for downloading the JAR web service is provided), dynamically allocated

The dynamical resources available to our experiment belong to the Italian Grid Infrastructure (IGI) (*http://www.italiangrid.org*), which is based on gLite, and to our local cluster, which is based on LSF.

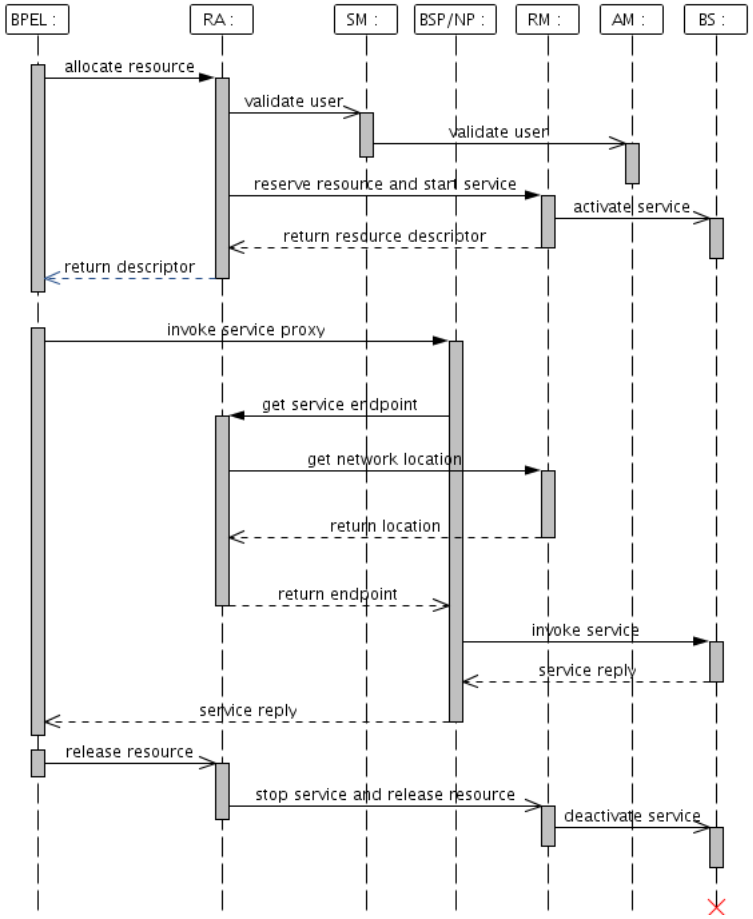

Figure 2. UML sequence diagram connecting BPEL instance, bearing and business services.

Two kind of users are considered here: (1) the user has a valid gLite key/certificate issued by a VO federated with IGI and wishes to use available IGI resources, (2) an anonymous user wishes to do some tests using the resources of our local LSF cluster. In both the cases the user is responsible for launching the visualization application on his/her desktop, and for providing input data such as classifier names, training and test dataset location, etc…

The anonymous user does not supply any credentials, while the Grid user is required to supply a valid proxy certificate. The input file is then sent to the BPEL engine, which creates a new workflow instance performing the following activities (for simplicity we are considering only the most important and leaving out the iteration over the attribute number):

1. invoke the RA providing the above resource list; the RA verifies the availability of the visualization service, prepares the data mining job and submits it to gLite middleware or to the RM responsible for LSF;
2. receive from the RA the resource availability notification (or an error) and the BSP endpoint to invoke for business service operations;

3. invoke data mining operations (which are routed transparently through the BSP), providing input data; BSP delivers the requests to the real service endpoint;
4. receive data mining service reply or notification with output data (notification is transparently routed through the NP);
5. invoke visualization service providing classifier accuracy to be viewed on user desktop;
6. release the resources: while the dynamically allocated data mining services are stopped and the resources freed, the statically allocated visualization service is left active.

CONCLUSION

This paper is a first attempt to bring together disparate e-Science resources (end-user, legacy and grid) and applications under the common umbrella of SOA and web services. Services organize their activity on the basis of both local and network information sources, and are related to a particular experimental context by a workflow describing the tasks to be executed and the context knowledge applied to solve a problem, to enact a decision or to achieve a goal. Even if the considered architecture in not exhaustive and implementation and usage scenarios are preliminary, the results are encouraging: it is possible to effectively run e-Science applications in an e-Science environment entirely built around business domain technologies such as web services and BPEL according to the SOA paradigm, while re-using existing infrastructures such as the Grid. Our prototype explored the adoption of Service Oriented Architecture to perform a distributed data mining experiment with the concurrent use of resources on the Italian Grid Infrastructure (heavy computation) and on the user desktop/laptop (visualization).

As a whole, the experiment is also a service with added value, and because of this it is potentially interesting for external organizations. It is possible to extend the proposed approach, both from the architectural point of view and on the implementation side, with new technologies in the area of Web applications by taking into account collaboration issues, support provided by open standards and the continuous evolution of scientific cooperative environments and computational platforms. Additional modular potentialities can be provided for tuning, recovery, and evolutionary characteristics.

In addition, the described architecture shares the objectives of Portals used for enabling distributed, federated subjects communicate toward a cooperative purpose, maintaining their local activities and autonomy, and, when needed and established on the basis of cooperative rules. One of the aspects to deepen is hence the organizational structure of the complex "business" policies necessary to execute the experiments and to correctly share data and results.

ACKNOWLEDGEMENTS

The authors acknowledge Cybersar Project and the Italian Grid Infrastructure (IGI) for the use of their computing facilities.

REFERENCES

Akram, A., Meredith D. & Allan, R. (2006). Evaluation of BPEL to Scientific Workflows. *Sixth IEEE International Symposium on Cluster Computing and the Grid, CCGRID'06* (pp. 269–274). IEEE Computer Society.

Bosin, A., Dessì, N., Madusudhanan, B., & Pes, B. (2010). Will SOA accommodate the next step of e-science? *10th Annual International Conference on New Technologies of Distributed Systems, NOTERE 2010* (pp. 303-308). IEEE Computer Society.

Bosin, A., Dessì, N., & Pes, B. (2011). Extending the SOA paradigm to e-Science environments. *Future Generation Computer Systems*, 27, 20-31.

Brandic, I., Pllana, S. & Benkner, S. (2006). High-level Composition of QoS-aware Grid Workflows: An Approach that Considers Location Affinity. *Workshop on Workflows in Support of Large-Scale Science, WORKS'06*. IEEE Computer Society.

Chao, K., Younas, M., Griffiths, N., Awan, I., Anane, R. & Tsai, C. (2004). Analysis of Grid Service Composition with BPEL4WS. *18th International Conference On Advanced Information Networking And Applications, AINA'04* (vol. 1, pp. 284-289). IEEE Computer Society.

De Roure, D., Gil, Y., & Hendler, J. A. (eds.) (2004). *Special Issue on E-Science*. *IEEE Intelligent Systems*, 19(1), IEEE Computer Society.

Deelman, E., Singh, G., Su, M., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A., Jacob, J. C. & Katz, D. S. (2005). Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems. *Scientific Programming Journal*, 13(3), 219-237.

Deelman, E., Gannon, D., Shields, M. & Taylor, I. (2009). Workflows and e-Science: An Overview of Workflow System Features. *Future Generation Computer Systems*, 25, 528-540.

Dörnemann, T., Friese, T., Herdt, S., Juhnke, E. & Freisleben, B. (2007). Grid Workflow Modeling Using Grid-Specific BPEL Extensions. *German e-Science Conference 2007, GES2007*. Karlsruhe.

Elmroth, E., Hernandez, F. & Tordsson, J. (2010). Three Fundamental Dimensions of Scientific Workflow Interoperability: Model of Computation, Language and Execution Environment. *Future Generation Computer Systems*, 26, 245-256.

Emmerich, W., Butchart, B., Chen, L., Wassermann, B. & Price, S. L. (2006). Grid Service Orchestration Using the Business Process Execution Language (BPEL). *Journal of Grid Computing*, 3(3-4), 283–304.

Fahringer, T., Prodan, R., Duan, R., Hofer, J. & Nadeem, F. (2007). ASKALON: A Development and Grid Computing Environment for Scientific Workflows. In Taylor, I. J.,

Deelman, E., Gannon, D. B. & Shields, M. (Eds.), *Workflows for eScience: Scientific Workflow for Grids* (pp. 450-471). Springer-Verlag.

Fox, G. & Ganno, D. (2006). *A Survey of the Role and Use of Web Services and Service Oriented Architectures in Scientific/Technical Grids* (Tech. Rep.). Indiana University.

Goble, C. A. & De Roure, D. (2007). myExperiment: social networking for workflow-using e-scientists. *2nd Workshop on Workflows in Support of Large-Scale Science, WORKS '07* (pp. 1-2). ACM.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 10-18.

Leymann, F. (2006). Choreography for the Grid: towards fitting BPEL to the resource framework. *Concurrency and Computation: Practice and Experience*, 18(10), 1201– 1217.

Mc Phillips, T., Bowers, S., Zinn D. & Ludascher B. (2009). Scientific Workflows for mere mortals. *Future Generation Computer Systems*, 25, 541-551.

Oinn, T., Li, P., Kell, D. B., Goble, C. & Goderis, A. (2007). Taverna / myGrid: aligning a workflow system with the life sciences community. In Taylor, I. J., Deelman, E., Gannon, D. B. & Shields, M. (Eds.), *Workflows for eScience: Scientific Workflow for Grids* (pp. 300-319). Springer-Verlag.

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-Oriented Computing: State of the Art and Research Challenges. *Computer*, Nov. 2007, 64-71.

Pennington, D. D., Higgins, D., Townsend Peterson, A., Jones, M. B. & Ludäscher, B. (2007). Ecological Niche Modeling Using the Kepler Workflow System. In Taylor, I. J., Deelman, E., Gannon, D. B. & Shields, M. (Eds.), *Workflows for eScience: Scientific Workflow for Grids* (pp. 91-108). Springer-Verlag.

Slominski, A. (2007). Adapting BPEL to Scientific Workflows. In Taylor, I. J., Deelman, E., Gannon, D. B. & Shields, M. (Eds.), *Workflows for eScience: Scientific Workflow for Grids* (pp. 208-226). Springer-Verlag.

Taylor, I., Shields, M., Wang, I. & Harrison, A. (2005). Visual Grid Workflow in Triana. *Journal of Grid Computing*, 3 (3-4), 153-169.

http://code.google.com/p/jlite, last accessed Feb, 2010

http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf, last accessed Feb, 2010

http://glite.web.cern.ch/glite, last accessed Feb, 2010

https://jax-ws.dev.java.net, last accessed Feb, 2010

http://utopia.cs.manchester.ac.uk, last accessed Feb, 2011

http://www.oasis-open.org/committees/wsbpel, last accessed Feb, 2010

http://www.platform.com/workload-management/high-performance-computing, last accessed Feb, 2010

http://www.sun.com/software/sge, last accessed Feb, 2010

http://www.globus.org, last accessed Feb, 2010

http://www.italiangrid.org, last accessed Feb, 2010

http://www.oracle.com/technetwork/java/javase/downloads/index.html, last accessed Feb, 2011

http://www.mygrid.org.uk, last accessed Feb, 2011

http://www.taverna.org.uk, last accessed Feb, 2011

http://www.mygrid.org.uk/tools/myexperiment, last accessed Feb, 2011

http://www.mygrid.org.uk/tools/biocatalogue, last accessed Feb, 2011

http://www.mygrid.org.uk/tools/e-labs, last accessed Feb, 2011