Mathematica
Balkanica

New Series Vol. 24, 2010, Fasc.3-4

# Services on Application Level in Grid for Scientific Calculations

*Radoslava Goranova*

The Grid is a hardware and software infrastructure that coordinates access to distribute computational and data resources, shared by different institutes, computational centres and organizations. The Open Grid Services Architecture (OGSA) describes an architecture for a service-oriented grid computing environment, based on Web service technologies, WSDL and SOAP. In this article we investigate possibilities for realization of business process composition in grid environment, based on OGSA standard.

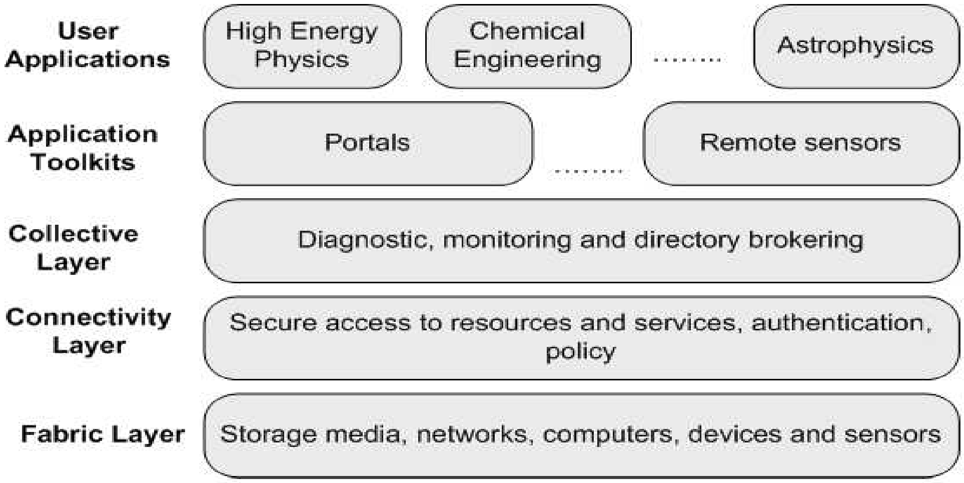
*AMS Subj. Classification*: 00-02, (General)

*Key Words*: SOA, Grid, Service orchestartion

## 1. **Introduction**

The computational Grid is a hardware and software infrastructure that coordinates access to distribute computational and data resources, shared in a large scale by different institutes, computational canters and organization. Grid resources can be storage systems, computers, software programs, even satellites and special devices. The key concept behind the Grid technology is the ability to agree the resource sharing among different participating parities (users or providers). The term sharing is used in most common sense. In Grid infrastructure sharing means direct access to computer, software, data and other recourses, not just file sharing. This sharing is coordinated by the resource provider and users who agreed about the way how the resources can be accessed. In order to organize users' rights, permission and security in the Grid, terms like virtual organizations and policies are used. Briefly, virtual organization is a set of individuals or institutions which share common goal and are spread all over the world.

From architectural point of view the Grid was defined by C. Kesselman and I. Foster in [1] as five layered protocol architecture. This architecture defines how the users can interact with the Grid resources. The five layered architecture consists of: fabric layer, connectivity layer, collective layer, application layer and user application layer.



The fabric layer provides resources - computational, storage systems, network resources, specific devices, sensors even clusters or pools. The connectivity layer defines communication and authentication protocols for exchanging data between fabric layer and secure mechanism for resource access. The collective layer contains protocols and services for resource interaction monitoring, diagnostics, scheduling, brokering, service discovering and others. The application layer provides software tools: application toolkits, API, SDK, portals and libraries for access to the services defined and implemented in collective layer. The user application layer covers user defined applications and user defined grid services.

## 2. Service-oriented Grid

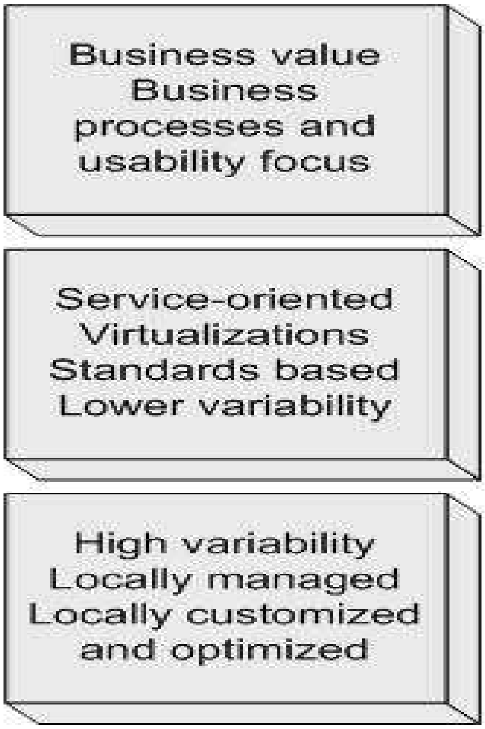
The concept of service-orientation is very well defined in [2]. The Service-Oriented Architecture (SOA) provides terms for formal description of the system, defining system's functions, properties and interfaces. A main unit in SOA is the service. Most commonly a service can be described as a software component that realizes specific system's logic and provides it to the end client by network access. Services possess the following characteristics:

(1) Services can be individually used or integrated in composition of services.
(2) Services communicate by message transfer.
(3) Services can be part from workflows.
(4) Services can be self-contained or they may depend on the availability of outer resources or other services.
(5) Service provides information for their interfaces, policies and communication they support.

The service-oriented architecture is very suitable for constricting complex distributed systems over heterogeneous environment, providing them with advantages as easy integration, flexibility and transparency. Service-oriented Grids follow the principles of service-orientation defined in SOA.

The Open Grid Services Architecture (OGSA) [3] describes an architecture for a service-oriented grid computing environment for business and scientific use. OGSA is based on Web service technologies, WSDL and SOAP. OGSA assure interoperability on heterogeneous systems so that different types of resources can communicate and share information.

OGSA represents three major logical and abstract tiers.



The first tier includes the basic resources as CPUs, memory, disks, licenses and OS processes. These resources are usually locally owned and managed and they can high variability in their characteristics, quality of service availability. The second tier represents a high level of virtualization and logical abstraction. The service-oriented architecture of OGSA implies that virtualized resources are represented as services and that interaction with them can be initialized by any service from the architecture. The services from this tier need to use and manage resources from the bottom tier in order to deliver capabilities of individual service. At the third tier are the applications that use the OGSA capabilities to realize user functions and processes.

OGSA realized this middle layer in term of services, the interfaces, the individual and collective state of resources and the interaction between them within a service-oriented architecture.

## 3. **Problems definition**

The domain of our research is system services on application level in service-oriented Grids. The goal is to investigate possibilities for realization of the business process composition in grid environment, based on the OGSA standard. By business process we mean a set of services ordered into scheme for execution. The main problem which has to be solved is to define a base service for business process orchestration or choreography in Grid.

OGSA defines in general the need of services composition, but does not specify in details how exactly these services will look like or how they will interact with the other components of the architecture. OGSA specification mentions that for services composition a variety of mechanisms like choreography, orchestration or workflow can be used, but also clarifies that OGSA will not define a new mechanism in this area and will rely on the existing standards for that. The proper definition of a service for service composition in the service-oriented Grid is tightly related with proper understanding of: orchestration, choreography and workflows.

In the next section we will describe these three terms and the difference between them and will try to prove that orchestration mechanism is the most suitable one in service-oriented grid environment.

## 4. Orchestration vs. Choreography

The ability of a system to provide mechanisms for building services composition is extremely important, especially for development of complex applications. The service-oriented systems possess this ability and it is called composability. In the way it is define in SOA, the composability of a system guarantees that services defined in the system can be composed in a more complex service and this service also can be part of another service. The process of the service composition is recursive.

According to that how the process is organized we can distinguish two types of service management: orchestration and choreography.

In orchestration, the process controls all contained services and coordinates execution of their operations. In orchestration we have centralized control, which is carried out by the service coordinator. The coordinator takes care for the execution order of containing services and their operations.

In choreography there is no centralized service coordinator. Instead, the process consists of a set of services which are equal. By equal we mean that every service holds information for the services with which it has to interact. All this is based on message exchange between services.

These definitions are general and are relative for every service-oriented system. However, our point of research is service-oriented grid systems, which have additional requirements which we have to conform with. In a service-oriented Grid environment we have to deal with the dynamic nature of grid resources, with grid security and with the indefiniteness of time for execution of the process.

As we already mentioned above the nature of the Grid is dynamic. We can not rely on the fact that the Grid resources will be permanent in the environment. The resources can appear or disappear, because of many reasons: power failure, network problems, service inaccessibility etc. We must assume that various parts of a services composition will fail. This is an issue, which we have to take in mind since we want to give a definition for service composition.

Another issue which we have to deal with is the indefiniteness of time for the process execution. Namely, because of the dynamic of the Grid we have to foresee the ability some of the Grid resources (services) to be temporary unavailable. The approach in that case is to replace the unavailable service with a similar one, which provides the same functionality. But we have to take into account, that although the service provides the same functionality, the time for task execution can be different.

Also, another critical issue is the ability to quickly bind the part of the process (task) to the appropriate Grid resources. And last, but not least we have to take into account all security issues that can arise during the execution of the process. For example if a grid user have rights to access one grid resources but does not have rights to access another grid resources and those resources are part of the process. The execution of this process most probably will fail due to security reasons.

If we conclude, management of service composition has to be service, which has to be well defined and described. Standards for description of service composition in Grid have to be extensible and to provide entities for exception handling.

The workflow is another, often mentioned term for service composition. In the terms of Grid, the workflow is defined by [4] as "The automation of the process, which involves the orchestration of a set of Grid services, agents and actors that must be combined together to solve a problem or to define a new service". Obviously, the workflow is a special case of orchestration, which concerns not only services but agents and actors. The actor is an external source, who influences the workflow. Or more concrete, the actor can be a person, who interrupts the workflow. The process can not continue without person's decision or person's choice.

The main dilemma which is the most suitable mechanism for a service-oriented grid environment, orchestration or choreography is still remaining. In order to answer this question, we will focus on existing standards for choreography and for orchestration - WS-CDL and WS-BPEL.

Historically viewed, for web services orchestration, the different services providers as IBM, BEA and Microsoft used different approaches. For example IBM used WSFL [5] language for business process description and Microsoft used XLANG [6]. In 2002, OASIS [7] defined a specification called BPEL4WS, which combined graphical process orientation from WSFL and the structure process construction from XLANG, in a new BPEL standard. The role of BPEL4WS [8] standard is to define new web services on the base of existing ones. It is a language for implementation of process orchestration, which is supported from the main web services vendors as IBM, BEA, Oracle and Microsoft.

The Web service business process execution language (WS-BPEL) is an XML based programming language for description of high level business processes. WS-BPEL provides methods for processes description and web services interaction. BPEL is the industry standard for orchestration, supported by the leading providers.

WS-BPEL provides methods for modelling business processes like: sequences, parallel threads, choices, external service invocations, exception handling and etc.

The Web Services Choreography Description Language (WS-CDL) [9] is an XML-based language that describes peer-to-peer collaborations of parties by defining, their common and complementary observable behaviour; where ordered message exchanges result in accomplishing a common business goal.

The WS-CDL model consists of entities, which describe interactions, like channels, participants, roles and work units and activities.

From grid point of view, WS-BPEL as a standard is a good candidate for service composition because of the following reasons:

(1) possibility language notation to be extended in order to meet the Grid requirements;
(2) the language specification is still evolving and supported
(3) the language provide mechanism for process execution, which is the main difference between WS-BPEL and WS-CDL

If we take into account, that WS-BPEL is a language for process execution and the reasons mentioned above we can logically conclude that orchestration as a mechanism is more suitable for grid environment, so far as it concerns existing web services standards.

## 5. **Conclusion and Future work**

In this article we describe two mechanisms for service compositions and choose a standard for formal description of the orchestration service, which will be most suitable in Grid environment. Still we have to specify the service, taking into account OGSA requirements and to deploy proposed Grid service into grid environment which implements OGSA.

## References

[1] I. Foster, K. Kesselman, *The Grid2: Blueprint for a New Computing Infrastructure*, 2004.

[2] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, 2005.

[3] I. Foster, Keshimoto, *The Open Grid Service Architecture, Version 1.5* [http://forge.gridforum.org/projects/ogsa-wg], 2006.

[4] G. Fox, D. Gannon, *Workflow in Grid System,* [http://grids.ucs.indiana.edu/ptliupages/publications/Workflow-overview.pdf], 2004.

[5] *Web Services Flow Language*, [http://xml.coverpages.org/wsfl.html].

[6] *XLANG,* [http://xml.coverpages.org/xlang.html].

[7] *OASIS,* [http://www.oasis-open.org/home/index.php].

[8] *WSBPEL,* [http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html].

[9] *WS-CDL,* [http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217].

Department of Computer Science,
Faculty of Mathematics and Informatics,
Sofia University ”St. Kliment Ohridski”,
5, James Baucher Str.
1164 Sofia, Bulgaria

E-Mail: `radoslava@fmi.uni-sofia.bg`