WSRF-BASED VIRTUALIZATION FOR MANUFACTURING RESOURCES

Lei Wu, Xiangxu Meng, Shijun Liu, Chenlei Yang, Xueqin Li
School of computer science and technology, Shandong University, Jinan, 250100, P.R.China

Keywords: Manufacturing Grid, manufacturing resources, resource virtualization, encapsulate template.

Abstract: Manufacturing Grid (MG) is a cooperation platform and resource sharing system among manufacturing enterprises. To be shared in MG, the manufacturing resources should be virtualized and exposed as web services. The paper presents a WSRF-based approach to virtualize resources conveniently in MG including many resource encapsulation templates and a resource container. The design principle, the implementation of the resource encapsulate template and the resource container are described in detail, where the resource container is a WSRF service deployed in GT java core and can expose resources as WS-Resource. Through these tools, resource providers can encapsulate their resources with the encapsulation template and deploy them in the resource container, then the resource will be virtualized and exposed as web service and can be shared in grid. A prototype platform to demonstrate the validity of the method is introduced at last.

1 INTRODUCTION

The principles of virtualization - the notion of liberating functionality and management functions from dedicated physical devices – have been around for decades. Virtualization technologies encompass a variety of mechanisms and techniques used to decouple the architecture and user-perceived behavior of hardware and software resources from their physical implementation. (Renato Figueiredo, 2005)Resource virtualization provides an abstraction that allows applications to access physical resources indirectly, without being tied directly to specific physical resources (such as servers or clusters). This virtualization software provides a common interface for applications. (N_emeth, 2003) points out that the essence of a Grid is the virtualization of resource and the virtualization of the concept of a user. Grids apply the concept of resource virtualization by aggregating numerous compute resources (from mainframes to desktops) and abstracting them into a single, unified resource that provides, in effect, a single virtual computer that can be accessed as needed or on demand.

The term of manufacturing grid (MG) is the applying of grid technologies on manufacturing, which is really a cooperation platform and resource sharing system among manufacturing enterprises.(Liu S.J,2005) Manufacturing resources are facilities involved in the entire product lifecycle, including material resource, information resource, technique resource, human resource, financial resource and so on. Some manufacturing resources can be shared through network easily, such as various information, data, and software, which can be used in cooperation work supported by MG. Software and computing service can be called via the network by means of Web services. But there are still many manufacturing resources which cannot be shared directly through network, such as materials, equipments and so on. (Liu S.J,2004).To be shared, these resources need to be described, expressed, encapsulated and deployed in MG. Via virtualization, all manufacturing resources are exposed as Web services, which eliminates the heterogeneousness of resources and shortens the distance between them.

Web Services are the technology of choice for Internet-based applications with loosely coupled clients and servers. That makes them the natural choice for building the next generation of grid-based applications. However, Web Services do have certain limitations. Plain Web services are usually stateless (even though, in theory, there is nothing in the Web Services Architecture that says they can't be stateful). This means that the Web service can't "remember" information, or keep state, from one invocation to another. So plain Web Services(as

Wu L., Meng X., Liu S., Yang C. and Li X. (2006).
WSRF-BASED VIRTUALIZATION FOR MANUFACTURING RESOURCES.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - SAIC*, pages 132-137
DOI: 10.5220/0002452301320137

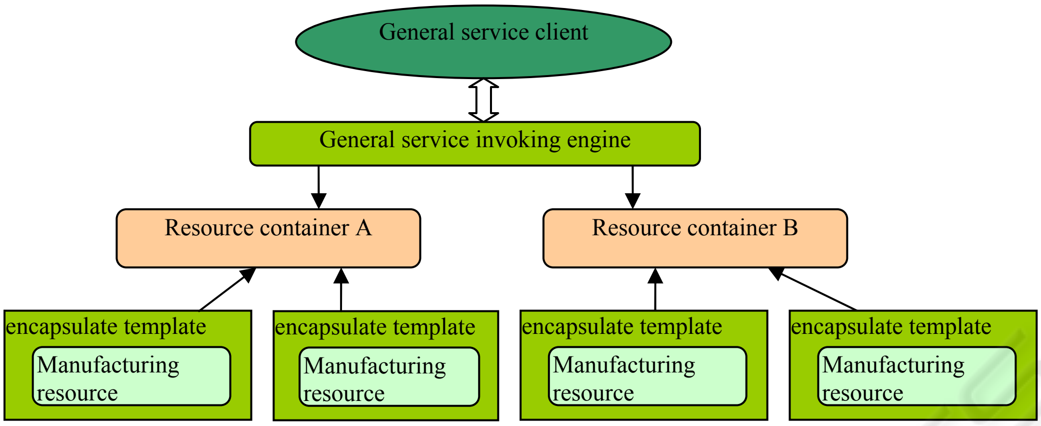


Figure 1: the structure of MG.

currently specified by the W3C) wouldn't be very helpful for building a grid application. However, Grid applications do generally require statefulness. So, we would ideally like our Web service to somehow keep state information (Borja Sotomayor,2004). The WSRF specifications were introduced in January 2004, which define the WS_Resource construct, a composition of a web service and a stateful resource described by an XML document (with known schema) that is associated with the web services porttype and addressed by a WS_Addressing Endpoint Reference(EPR) [IBM, BEA and Microsoft]. The four WSRF specifications being standardized in OASIS[OASIS] define how to represent ,access, manage, and group WS-Resources.

(Rajaram Shivram Kumar,2004) adopts Web Services Resource Framework (WSRF) as a major mechanism for virtualization. Using WSRF, manufacturing resource will be exposed as a WS-Resource. However, this method has certain limitations. Every resource must be virtualized as a separate web service, which makes the management of these services very complex. Meanwhile, most of resource providers can't hold the WSRF specifications, so it is difficult for them to virtualize their resources and deploy them in grid. We present the concept of resource encapsulation template and resource container. The resource container is actually a WSRF service, which can adopt resources implemented by Java, C and .NET language and virtualized them as WSRF services.

The rest of this paper is organized as follows. We first introduce the design principle of our approach in Section 2. In Section 3 the resource encapsulation template is presented. The resource container is described in detail in Section 4. The prototype system is outlined in section 5. Finally the Section 6 concludes the paper and outlines some future work.

2 DESIGN PRINCIPLE OF RESOURCE VIRTUALIZATION

To be shared in MG, manufacturing resources need be virtualized as web services. There are many requirements as follows:

(1)Resource virtualization should make various manufacturing resources be shared conveniently in MG.

(2)Resource virtualization should make resources offer their functionality absolutely, meanwhile, hide the implementation detail.

(3)Resource virtualization should make the using and management of resources more convenient.

(4)Resource virtualization should make the searching of resource more easy and exact.

There are various virtualization methods. One way is encapsulating resources as web services. Manufacturing resources are different in express form, management mode, sharing mode, using approach, trade mode, and so on. So it is difficult to manage and use the resources encapsulated without uniform specification. According to above requirements several resource encapsulation templates are presented, which are the specification and certain interface followed when encapsulating resources' logical function. It supports java, c and .net program language. Resource providers can encapsulate their resources using these templates. To expose these resources encapsulated with our template as web service and shared in grid, we design a resource container, which is actually a WSRF service deployed in GT service container. It

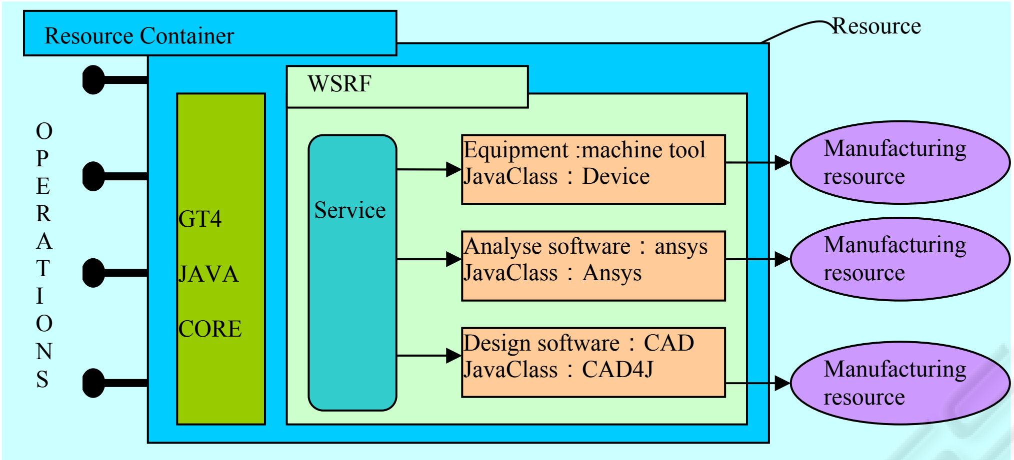


Figure 2: Architecture of Resource Container.

can adopt resources encapsulated with the template presented above and automatically virtualized them as WSRF services. So resource providers can virtualize their resources as web services through implementing some interface. They can use some common program language such as Java, C and .NET. The resource encapsulate template and the resource container will be described in detail in below sections. The structure of the grid is shown in Figure 1. Manufacturing resources are encapsulated with the resource encapsulation template and added in one of resource containers. Resource container can be deployed in any node of the grid. There are two resource containers in our prototype. A general service invoking engine is developed, which can invoke resources deployed in different resource containers to complete the users' task.

3 RESOURCE ENCAPSULATION TEMPLATE

Resource encapsulation template is some encapsulation specifications and certain interface that resource providers should implement. The resource container supplies resources encapsulated by diverse encapsulation templates based Java, C and .NET. A template based java is presented in detail in this paper. When resource providers encapsulate their resources, they must implement the interface as follows.

Public interface IMGResource{

Public void initResource(IResourceRefrence ref);
// this method is used to initialize the resource, 'ref' is the reference of the resource .the resource state will be set in this method

Public HashMap getTaskInputParameters();
// return the input parameters of the task.

Public HashMap executeTask (HashMap input) ;

// execute a task. All the logical function that the resource offer is implemented in this method, the result will be return in HashMap structure.

Public file getResDescription ( ) ;

// get some description of the resource .we set xml schema for every kind of resource ,the description of the resource will be described with xml document .

}

Resource providers implement the above interface to encapsulate the resources' logic functions and deploy them in the resource container, and then the resources can be exposed as WSRF services and offer their functionality in a standard Web service environment to be shared in grid.

4 RESOURCE CONTAINER

To expose resources encapsulated with our template as web services, resource container is designed and developed.

4.1 Design Principle of the Resource Container

The resource container is actually a WSRF service. It can be deployed in GT service container conveniently. It can adopt resources encapsulated by above template as it's resources and expose them as web services shared in grid.

The architecture of the resource container is shown in Figure2.

The encapsulated resources are deployed in the resource container, at the same time the information of the resource including the address of the resource container and the resourceID will be registered in a private UDDI. The registry gets the description of the resource, which is a facility for resource search.

A general resource invoking engine is designed in the job manager, which can invoke resources deployed in the resource container. When users want to use some resources in grid, they firstly search the resources by keywords from private UDDI registry. Resources satisfying the requirement will be offered by the service invoking engine. Then the invoking engine will invoke the selected service to complete the task.

4.2 How the Container Support Resource Virtualizing

(1) Hot resource deploy

The encapsulated resources can be hot deployed in the resource container. The resources can be used without stopping the server container. If the container has to restart to use the new resource, the task running will be break in, which is not allowed. So hot resource deploying is very significative.

(2) Remote resource deploy

The remote deploying interface is provided, the resource providers can encapsulate their resources with the resource encapsulation template, deploy them in resource container and register into MG from portal, which make the resource providers virtualize and deploy their resources by themselves conveniently

(3) Easily deployed and used

The resource container is actually WSRF service, which can be easily deployed in some domain of the grid with the "globus-deploy-gar" command. It doesn't need peculiar configuration. All resources in one resource container can be deployed in another resource container as a whole.

(4) Hiding implementing information

The encapsulated resources hide the implementation information of the resources and only the function interface is exposed to users, which increases the security of resources.

(5) Self-describing of resource

Different xml schemas are defined for every kind of resources. Resources are described with the xml document. Resources' description information can be set when they are initialized.

(6) Effectively monitor

The resource container provides the getResState() interface ,which ensures that the application's state and other data available .thus job invoking engine can monitor running applications to be able to fulfill an accepted SLA.

4.3 Implementation of the Resource Container

There are many difficulties in designing the resource container. This section describes the particular design of the resource container in detail. Resource container is a WSRF service, resources encapsulated with above template will be added into the resource container as WS-Resource. The particular design of the resource container will be described in detail in the next part of the section.

(1) Resource structure

```
MGRes{
    String type;
    String resName；
    String className,
    String resState；
    IMGResource aResource；
}
```

The className is the name of the resource implementation class, which must implement the provided interface. 'resState' is the state of the resource, which is set by the resource implement class. The resource invoking engine can monitor the resource by getting this attribute. 'aResource' is the reference of the resource logical function. It points to the resource implement class.

(2) Permanence of the resource

The information of the common WS-Resource will be lost when the service container restart. To store this information enduringly, the paper describes the way on serializing resource. The resource information will be saved in files. When the service container start, the file will be read to unserialize the resources, meanwhile the resource will be initialized. The state of the resource will be reseted. The resource file will be modified when resources are added, removed and modified. Only the type, resName and className will be serialized in file. The 'resState' and 'aResource' need to be set when the resource is initialized.

(3) Dynamic Load

When the resource is serialized, 'aResource' is not serialized. So the implement class must be dynamicly loaded and set the 'aResource', make it point to the implement class.

The dynamic load program as follows：

```
try {
IMgResourceImpl myObj=null;
Class myClass=null;

// get Jar file and classname
jarFileDir=C:/ws-core-4.0.0/bin/Temp;
jarFileName=" MgResourceImpl.jar";
String jar=jarFileDir+"/"+jarFileName;

// Create URL for URLClassLoader
URL[] urls = {new URL("file:///" +
jar)};

// get the ClassLoader
ClassLoader
cl=Thread.currentThread().getContextCla
ssLoader();
URLClassLoader classLoader = new
URLClassLoader(urls,cl);

// Find the class
myClass =
classLoader.loadClass(className);

// Create an instance of the driver
myObj=
(IMgResourceImpl)myClass.newInstance();

resourcePackageResource.SetAResource(my
Obj);

//initialize the resource refrence.
ref=(IResourceRefrence)resourcePackageR
esource;
myObj.initResource(ref);
}catch(Exception e){
e.printStackTrace();
}
```

5 PROTOTYPE

We have developed the middleware to support the establishing of a manufacturing grid system. A prototype platform has been deployed to validate our idea. The portal (www.mgrid.cn) can be visited now. (Figure 3) Resource containers were deployed in two nodes. The resource provider can download the resource encapsulation templates from the portal, encapsulate their resources and deploy them in the resource container in the portal. Meanwhile the virtualized resources are registered in a private UDDI registry in the MGRID. The deploying and register step is shown in the Figure 4. Then the resource can be shared in grid.

Figure 3: the portal of MGRID.

A kind of CAE software were virtualized with above resource encapsulation template and deployed in the resource container of our MGRID. Resource users can request this service through a invoking form from portal. Figure 5 shows the form user should fill out.

6 CONCLUSIONS

Virtualization of manufacturing resources and exposing them as web services is an ideal method of manufacturing resources sharing. This paper presents a virtualization method based on WSRF. The resource encapsulation templates based java and the resource container were introduced and developed. Resource providers can virtualize their resources as WSRF services conveniently and deploy them in network. A prototype has been developed to validate our idea. Various manufacturing resources have been virtualized and shared conveniently in our MG system now.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support provided for the Science & Technology Development Projects of Shandong Province (031110147，2004GG1104011, 2004GG1104017)

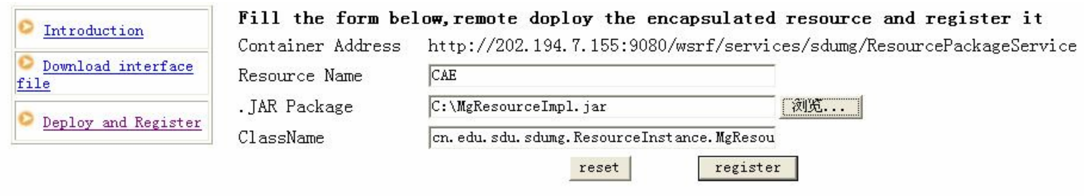
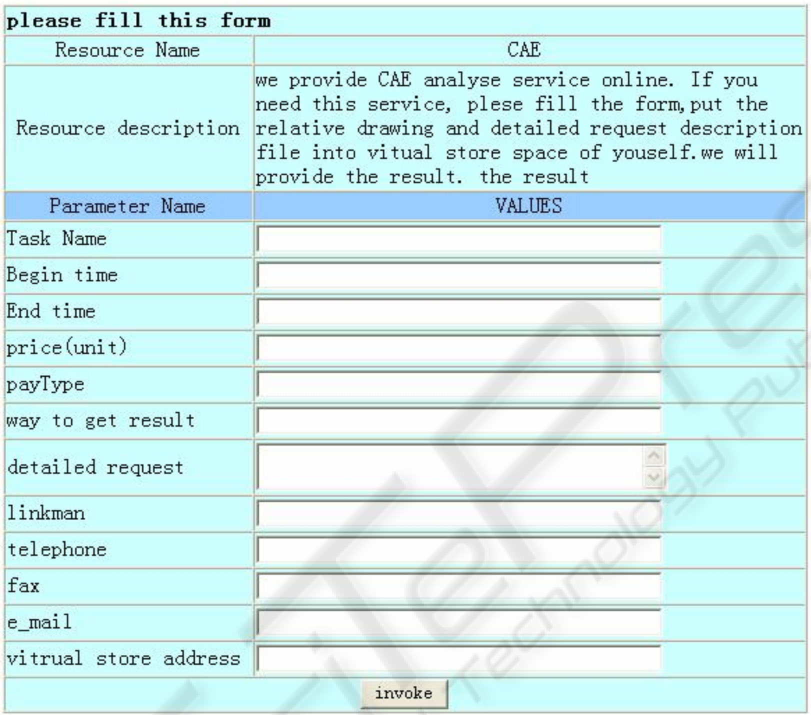


Figure 4: resource deploying and register.



please fill this form	
Resource Name	CAE
Resource description	we provide CAE analyse service online. If you need this service, plese fill the form,put the relative drawing and detailed request description file into vitual store space of youself.we will provide the result. the result
Parameter Name	VALUES
Task Name	
Begin time	
End time	
price(unit)	
payType	
way to get result	
detailed request	
linkman	
telephone	
fax	
e_mail	
vitrual store address	
invoke	

Figure 5: the form of the CAE service.

,Natural Science Foundation of Shandong Province(Y2004G08), National Natural Science Foundation of China(90612021)and the Research Fund for the Dectoral Program of Higher Education(20050422019).

REFERENCES

Renato Figueiredo, 2005. Resource Virtualization Renaissance. In *Computer Volume 38 , Issue 5 p28 - 31 .*

N_emeth, Z., Sunderam, V,2003. Characterizing Grids: Attributes, Definitions, and Formalisms. *Journal of Grid Computing*

Liu S.J, Meng X.X, Tong Y.X , 2005. SOMG: A Service Oriented Manufacturing Grid System. In *the Fifth International Conference on Computer and Information Technology*.CIT'05 Press.

Liu, S.J, Meng, X.X, Gong, B, 2004. Resource Modeling in Manufacturing Grid. In *the 11th ISPE International Conference on Concurrent Engineering: Research and Applications.*

Borja Sotomayor, 2004. The Globus Toolkit 4 Programmer's Tutorial University of Deusto.

IBM, BEA and Microsoft. 2004. WS_Addressig *http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobuspec/html/ws-addressing.asp*

David Snelling, 2004. OASIS Web Services Resource Framework(WSRF) TC *.http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf*

Rajaram Shivram Kuma, 2005.Virtualization for Manufacturing Web Services: a WS-RF approach .In International journal of information technology Volume 11, Number 3