A WS-Resource for Networked Instrumentation

Jungwook Seo , Ernie W. Hill
Dept. of Computer Science,
Univ. of Manchester
{seoj, e.w.hill}@cs.man.ac.uk

Ja Yl Ku
Seoul Korea
drkoo2@freechal.com

Abstract

We present a stateful Web Service based on the WS-Resource Framework (WSRF) using Globus Toolkit 4(GT4) Core, XML based languages and Java. The Web Service that we have implemented is designed particularly for a Scanning Electron Microscope (SEM) and is a transient and stateful Web Service, which can dynamically cope with requests from a number of clients. This paper proposes a solid method of the authorized access to real resources in the new emerging WSRF by the efficient use of Resource Properties with Resource Keys. We describe not only how the Resource Home manages WS-Resources but also how WS-Resources are activated and deactivated to allow clients to occupy instruments with resource keys. Additionally, a local job manager in the backend server is implemented for scheduling and executing job so as to integrate GT 4 with the existing instrument application. We show that this Web Service might be applied to a wide range of instrument resources as well as application servers and also provides various advantages with regard to the efficient use of instruments for collaboration over networks.

1. Introduction

"Grid Computing" [1] has emerged as the next generation of Internet in terms of large-scale resource sharing even though resources are distributed and owned by different multiple organizations across many sites. Resources can be defined as anything related to computing such as software, computers, instruments, networks, databases and so on. With respect to those resources, this paper addresses particularly instrument resources sharing with respect to implementing a stateful Web Service for a Scanning Electron Microscope (SEM).

Open Grid Services Architecture (OGSA) [2] announced by the Globus Grid Forum defines an open architecture for grid enabled applications for stateful services and the Open Grid Service Infrastructure (OGSI) [3] details the specifications of OGSA framework defining mechanisms for creating, managing, and exchanging information with respect to Grid services. However, OGSI does not interact well with existing Web Services so that the Web Services Resource Framework (WSRF) [5][6] has been developed by OASIS [5]. WSRF is a joint effort by the Grid and Web Services communities and addresses statefulness of Web Services, along with adding a number of other new features and standards. GT4 [4] includes a complete implementation of WSRF specifications used in this project, including WS-ResourceProperties [7], WS-Addressing [10], WS-ResourceLifetime [8], and WS-Notification [9].

On top of GT4 WSRF implementation, we have implemented a WS-Resource for the SEM such that a client can create a transient "SEM Resource" instance from the "SEM Resource Home" and keep specific job states in the instance as long as he or she wants to use it. An administrator must place a specimen into the specimen chamber of the SEM and ensure the SEM chamber is closed ready for job execution. According to the prepared specimen, the administrator will activate one of WS-Resources that have been registered when each client creates a SEM Resource instance. Recently, SEMs are fully operated by applications without manual operations. Therefore, all jobs will be executed in an automatic way supported by the SEM application without help from an administrator, in order to capture a specific SEM image. Consequently, this WS-Resource for the SEM will give some significant advantages including:

- Maximized use of instruments: When local users do not use their instruments, authenticated and authorized Grid users might be allowed to use them.
- Collaborations by virtual laboratories: a number of scientists or engineers can simultaneously share instruments at their home laboratories
- Reducing contamination of clean rooms in which SEMs are located and operated by enabling many users to work out of the clean room environment. Thus this system can reduce costs in maintaining clean rooms.

2. Reasons for WS-Resource

We have adopted WS-Resource (GT4) with some reasons. One of the important reasons is security, which is a very important factor for instrument servers. Current web security mechanisms are not enough to secure our instrument server. However, the security mechanism of the Globus Toolkit is based on the x.509 Public Key Infrastructure (PKI) certificates [11], using digital signatures. It is sufficiently reliable so as to be more suitable for securing our server. Another reason is that instruments can dynamically and flexibly be joined with any Virtual Organization (VO) [2], which is composed for transient purposes and will be seen more and more in the near future. For example "A cancer project" and "A magnetic thin film project". The most important reason is the capability of keeping the job status, which has not been seen in the current client server architecture on the web. This stateful feature is very important particularly to operating a SEM. For example, when we navigate the specimen stage to the next position using the relative XY coordinates, it is impossible without knowledge of the previous status. For these reasons, WS-Resource has been adopted to implement our instrument server.

3. Implementation

3.1 Overview

We have implemented a prototype WS-Resource for the SEM based on GT4. Two servers have been implemented: one is the front-end server for a hosting environment and another is the existing back-end server on which the existing SEM application is running as shown in Figure 1. A simple Local Job Manager has been implemented on the back-end server in order to integrate the GT4 with the existing SEM application.

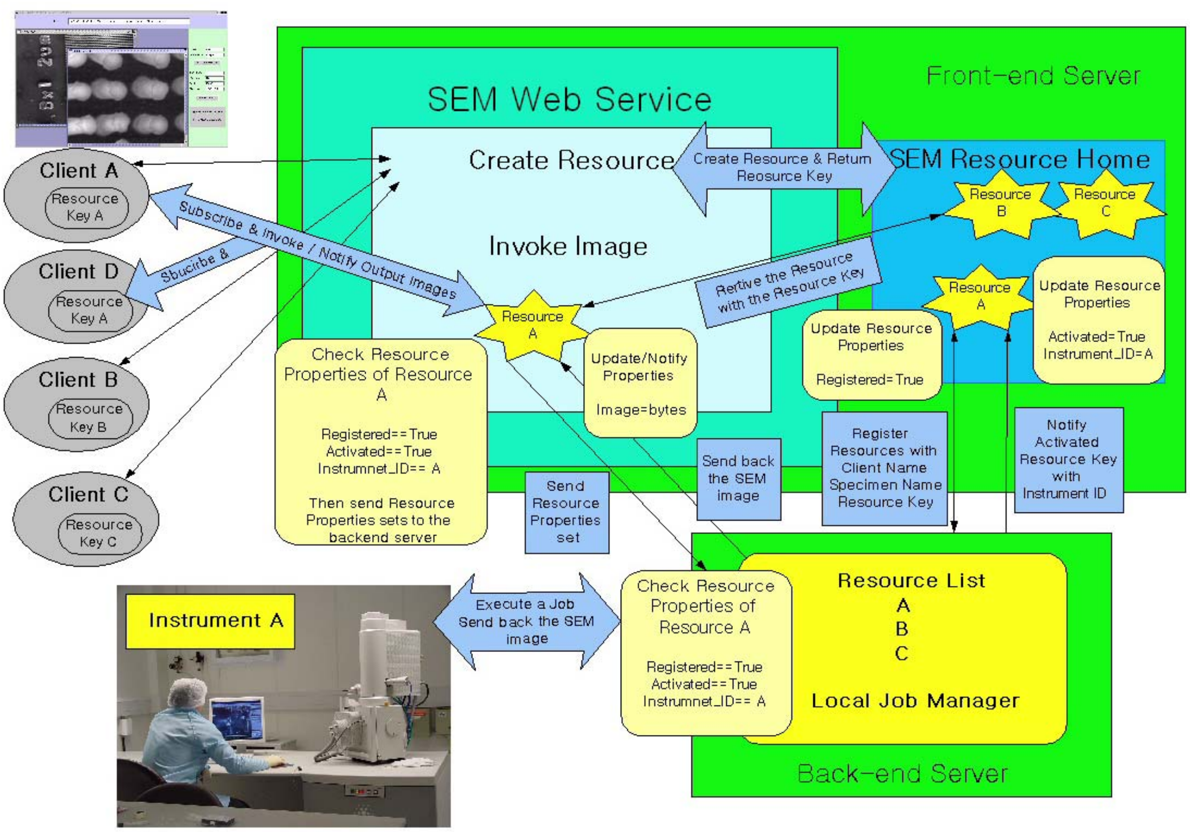


Figure. 1 The structure of SEM WS-Resource

Basically, communication between clients and the Web Service is established by Simple Object Access Protocol (SOAP) binding stubs [12] and between servers by java object streams. For security, Clients must have valid certificates signed by the Certification Authorities (CA) and must be registered on our access control list. However, we do not address any WS security mechanisms such as 'transport layer security' and 'message level security' in this paper.

3.2 Creating SEM Resources and Activation

Each client invokes creating a specific SEM Resource to the SEM Web Service with parameters such as 'Client Name' and 'Specimen Name'. The Web Service retrieves the SEM Resource Home (SRH), which creates a unique SEM Resource (SR) with a unique Resource Key (RK) using the hash function. When the SR is created, a set of SEM Resource Properties (SRP) of the SR will also be created as shown in Figure 2. Then, the endpoint reference will be sent back to the client. After creating each SR, SRH will register the SR with the SRP to the local job manager on the backend server and will also listen to notifications from the back end server to set activation or deactivation upon a particular SR. Therefore, an administrator who physically places the specimen under the SEM is able to determine which specimen belongs to which client. The decision will be notified to the SRH with both the RK of the selected SR and an instrument ID in order to activate the SR for the specific client. The SRH will then find the SR with the notified RK from the administrator. One of SRPs, 'Activated' will be set to true and another property 'instrument_ID' will also be set to 'A' in the SR. However, SRH will also keep the notified activated RK for a particular instrument (A). When any client invokes an SEM image to the Web Service, the RK of the client will be verified with the activated RKs of the SRH and the activated property of the retrieved SR must be 'true'. The local job manager on the back end server will also keep the activated RK so as to ensure that each job is for the activated SR as shown in Figure 1.

3.3 Job Execution

The activated client can access to the SEM and can execute jobs as long as the corresponding SR is activated. Each job will be executed by an interface (wrapper class) which is coded by the same sequence of the local user operation. For example, moving stage to the point ⇒ auto stigmator ⇒ auto focus ⇒ auto contrast control ⇒ capturing the image ⇒ sending the image back to the front end server. The image quality of this automatic way is not equivalent to that of local operations particularly in focusing the images. Clients can have even highly magnified SEM images ranged from 200 to 200000 in magnification so that they can have nano-scaled images according to three parameters such as 'position X' (μm), 'position Y' (μm) and 'Magnification' (200-200,000) which are part of SRP.

```
<xsd:element name="SemResourceProperties">
  <xsd:complexType>
    <xsd:sequence>
          <xsd:element ref="tns:Registered" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:Activated" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:InstrumentID" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:ClientName" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:SpecimenName" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:CurrentX" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:CurrentY" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:Coordination" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:Magnification" minOccurs="1" maxOccurs="1"/>
          <xsd:element ref="tns:OutputImage" minOccurs="1" maxOccurs="1"/>
          <xsd:element name="resourceKey" type="xsd:anyType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

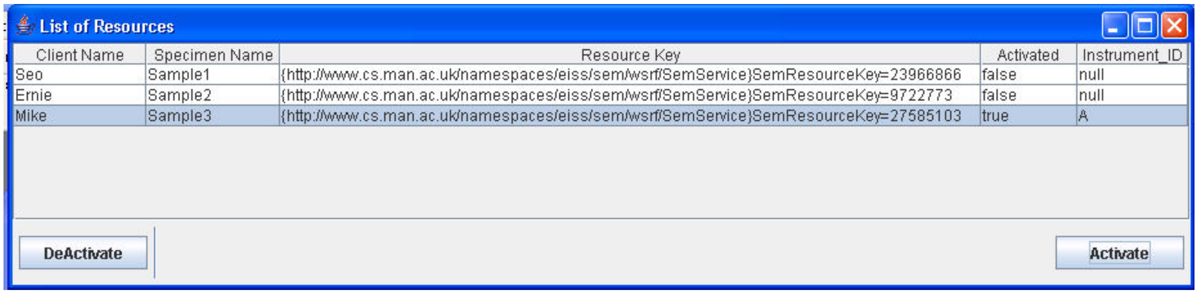
Figure 2. SEM Resource Properties (SRP)

If the client sets the relative xy coordinates in the parameter, the xy positions will be recalculated with the current xy positions of the SR properties. This is one of advantages of the stateful WS–Resource, which have not been seen in plain Web Services.

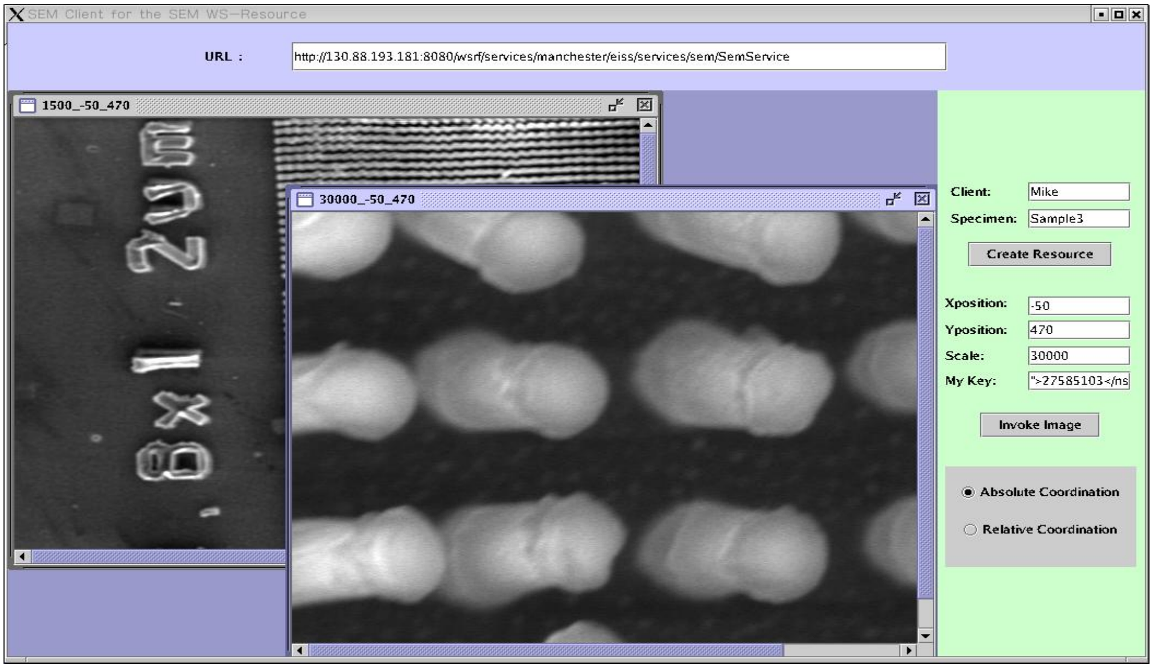
In Figure 1, Client A and D can share the endpoint reference [10] and subscribe an output image as a WS-Topic. Then, whenever an output image is updated in the SRP, it will be notified to all subscribers according to WS-BaseNotification [13].

3.4 Deactivation

The possession of the SEM by a client can be deactivated not only by the administrator on the backed server but also by the client according to WS-LifetimeResource. The administrator can deactivate the activated RK of the backend and it will be notified to the SRH. Therefore, SRH will set up its activated RK as null and retrieve the SR with the notified RK and set up the Activated property of the SR as false. On the other hand, the client can finish possessing the SEM by instant destroying the SR, which has kept jobs for the client. SHR will also set up its activated RK as null and notify to the back end server before destroying the SR.



(b) GUI for the administrator



(a) A Graphic User Interface for a client

Figure 3. The GUI of SEM WS-Resource

As a result, in both cases the client cannot access to the SEM any more. However, SRH will continue to listen to a notification for a new activated RK from the administrator on the backend server.

4. Results

We have tested our stateful Web Service in a prototype with a particular specimen. Figure 3 shows the output images in certain conditions. In Figure 3(a), the client named Mike creates a SEM Resource and gets back the Resource Key (27585103) as the "My Key" field. After the administrator activates Mike's Resource Key with an instrument id as shown in Figure 3(b), he can access to the backend server and start jobs. The backward image that is micro scaled is generated by parameters (Xposition: -45, Yposition: 470, Magnification: 1500). It shows a slightly better quality than the front nano-scale image. The quality of images entirely depends on the auto functions of the SEM application.

However, it takes considerable time to get an image around 20 seconds with a network speed of 100Mbps. The output image (712X484 pixels) generated by the SEM is compressed in the JPEG format to minimize size problem so that the size of each image is less than 0.3 Mega bytes. The 20 seconds delay mainly resulted from the SEM's automatic operating time for capturing an image rather than network delay. Moving the stage to a point takes 5seconds and auto focusing takes 8 seconds. Limitations to the SEM remote operation are required in order to avoid crashes between the SEM objective and specimens because the working distance between the lens and the specimen is generally 2-5mm for high magnifications. Therefore, we need to limit the range of input parameters depending on the shape of specimens and on the elevation of the specimen surface.

5. Future work & Conclusion

The main future work will focus on enhancing security for the stateful WS-Resource in each message exchange with WS-Security [14] and WS-SecureConversation [15]. For restricting the range of parameters in remote operating environments, the use of 3D virtual specimens is considered. Remote clients will initially receive a 3D virtual specimen and mainly work with the 3D virtual specimen in their sites. The main aim of the virtual specimen is to give general information about real specimens and to restrict the remote users' environment so that they are able to download real SEM images only for specific points of the virtual specimen when they found regions of interest. The main aim of this approach is to reduce interactions between remote clients and servers to avoid potential crashes.

We have shown that our WS-Resource presents benefits in terms of sharing instrument resources. Each client can keep the state of jobs in the SEM Resource with the unique Resource Key. We describe also how the SEM Resource Home that keeps SEM Resources interacts with the backend Server by both activating and deactivating Resource Keys for the authorized access to instrument resources. However, this authorization method can be applied to a wide range of existing resources. As scientific collaborations are becoming more geographical dispersed, this remote instrumentation through stateful Web Services will provide scientists with easy access to remote instruments without the need to be physical present at the instrument site, thereby saving the cost and time of research.

6. References

[1] Computational Grids: I. Foster, and C. Kesselman. (eds.). The Grid: Blueprint for a New ComputingInfrastructure. Morgan Kaufmann, 1999.

[2] The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. I. Foster, C. Kesselman, J. Nick, S. Tuecke, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[3] Open Grid Services Infrastructure (OGSI) Version 1.0. S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, D. Snelling; Global Grid Forum Draft

[4] www.globus.org

[5] www.oasis-open.org

[6] Modeling Stateful Resources in Web Services. http://www-106.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf

[7] WS-ResourceProperties http://www-106.ibm.com/developerworks/ library/ws-resource/ws-resourceproperties.pdf

[8] WS-ResourceLifetime http://www-106.ibm.com/developerworks/ library/ws-resource/ws-resourcelifetime.pdf

[9] WS-Notification family of specifications (WS-BaseNotification, WS-Topics and WSBrokeredNotification). http://www-106.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf

[10] WS-Addressing
http://www.ibm.com/developerworks/webservices/library/
ws-add/

[11] Gasser, M. and McDermott, E., An Tuecke, S., Engert, D., Foster, I., Thompson, M., Pearlman, L. and Kesselman, C. Internet X.509 Public Key Infrastructure Proxy Certificate Profile, IETF, 2001.

[12] The fundamental message enveloping mechanism in Web services. http://www.w3.org/TR/SOAP.

[13] WS-BaseNotification
ftp://www6.software.ibm.com/software/developer/library/ws
-notification/WSBaseN.pdf

[14]WS-Security
http://www-
28.ibm.com/developerworks/webservices/library/
ws-secure/

[15] WS-SecureConversation
ftp://www6.software.ibm.com/software/developer/library/ws
-secureconversation.pdf