Integrating Digital Libraries by CORBA, XML and Servlet[§]

Wing Hang Cheung, Michael R. Lyu, and Kam Wing Ng
The Computer Science and Engineering Department, The Chinese University of Hong Kong, Hong Kong
Email: {whcheung, lyu, kwng}@cse.cuhk.edu.hk

ABSTRACT

In this paper, we describe how we use a mediator-based architecture for integrating digital libraries. We discuss how we tackle the obstacles of firewalls in the expansion of our system by using XML and Java Servlet, which are used to achieve CORBA general communications and callback features across the firewalls.

1. INTEGRATING DIGITAL LIBRARIES

Nowadays, there is a trend to integrate several digital libraries to offer richer information. Here, we use the mediator architecture to integrate multiple digital libraries via the Internet. Mediators forward the client queries to the appropriate digital libraries or mediators, and then integrate the returned answers and forward them back to the clients. Meanwhile, the use of firewalls in the Internet leads to a communication problem. Many digital libraries are implemented in CORBA and use Internet InterORB Protocol (IIOP) to communicate, but IIOP is blocked by common firewalls. Though there are firewalls dedicated for CORBA [1,2], they are generally vendor-dependent and may not support certain CORBA features like callbacks. Elenko and Reinertsen [3] suggested a communication perspective of the cooperation between XML and CORBA. XIOP [4] is another proposed substitute of IIOP by XML data. Hence, we have developed a simple mechanism of using HTTP, XML and Servlet for tunneling through the firewalls and support different CORBA IIOP calls in a generic way.

2. SUPPORTING CORBA CALLS

In our system, we use a new object, named *MediatorGateway* to serve the purpose of connecting to another object in a server enclave behind a firewall. It implements the same interface as the *Mediator* objects and converts all the necessary parameters into an XML message, and sends it to the target mediator by HTTP calls. In Figure 1, Mediator *M* wants to call Mediator *SM*, which is merged with a Servlet component. *M* now calls the shadow object *MediatorGateway G*, and *G* converts the queries and parameters into an XML message and sends it to *SM* by HTTP calls going through the firewall. *SM* further makes queries to digital libraries in the server enclave and returns the answers, in XML format, to *G* by HTTP calls. *G* returns the answers to *M* by ordinary IIOP calls. In this case, *M* just performs a normal call and the external object reference is transparent to *M*. From the viewpoint of the objects in the client enclave, the *MediatorGateway* object inside the same client enclave is just a shadow object of the target *Mediator* object in the server enclave.

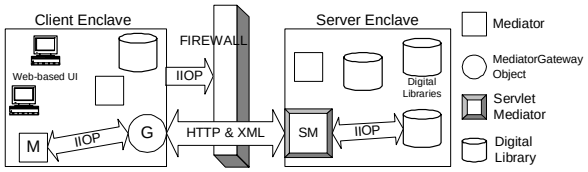

Figure 1. Using XML, HTTP & Servlet to support IIOP

To better help the users in getting information, we allow users to subscribe to various information topics, such that the system can inform them immediately when there is an update. This requires callbacks. In Figure 2, the client first gets a reference to the shadow server in the client-side enclave. Then the shadow server creates a Servlet component immediately in the client enclave for the server side to callback. The shadow server sends information, such as IP address and port number, of the newly created Servlet to server side. At the server enclave, a shadow client is created automatically at that time. The server uses an array to store all shadow clients that require callbacks. Whenever the server is updated, it calls the shadow clients to notify the client objects.

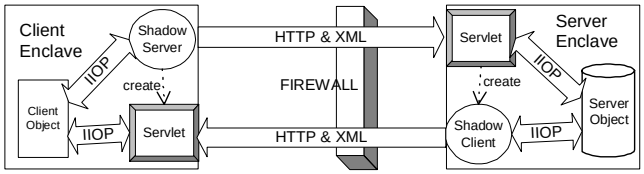

Figure 2. Our Callbacks mechanism

3. CONCLUSION

Our approach solves the incompatible IIOP firewall problem while security can still be maintained, as objects outside the CORBA enclave can only call the objects integrated with Servlet. System transparency is increased, as internal objects notice no difference in calling the target object and the shadow *Gateway*. System interoperability is enhanced, as using XML can well represent the transmitted messages. The system is not limited to CORBA objects because as long as the syntax of transmission messages is agreed, any kind of implementations can be used in clients and servers. Our design also has some weakness as HTTP calls is slower than IIOP, and we need extra workload to initialize and use the Java Servlet objects. Nevertheless, the overhead time is negligible comparing with average Internet access delay.

4. REFERENCES

[1] IONA Technologies. *Orbix Wonderwall Administrator's Guide*, June 1999.

[2] Visigenic Software, Inc. *Visigenic Gatekeeper Guide, version 3.2 edition*, February 1998.

[3] Mark Elenko and Mike Reinertsen. XML & CORBA. Application development trends, September 1999.
[4] Homepage of XIOP. http://xiop.sourceforge.net

§ The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. CUHK4193/00E).