

BY RICHARD B. VIPOND

# Legacy Systems Enablement: Part IV – Integration With World Wide Web (WWW) Servers

This article examines the WWW servers in this site's open systems development environment model as well as the software configuration management software that will create the global development environment.

**THE** goal of this four-part series is to demonstrate how to create an open systems development environment model (see Figure 1) from which a company can more readily migrate into a fully functional object-oriented environment such as CORBA.

Our model consists of two major areas of activity: the development environment and controlling software configuration software (SCM) software, which were explained in the previous articles of this series (Part I, *Technical Support*, July 1998; Part II, August 1998; and Part III, October 1998). The WWW servers as well as the multitude of associated gateways, emulators, databases and middleware servers are the supporting infrastructure that allow us access to our important legacy system assets.

This article examines the WWW servers in our open systems development environment model and takes a closer look at the SCM software. It also examines how important current leading edge development concepts are in relation to the concept of a global development environment. I will also discuss how a product like IBM's Component Broker, which manages the CORBA environment, will ultimately manage the global development environment.

## THE SERVER ENVIRONMENT

Our open development environment consists of numerous tools, technologies and standards that must conform to the fundamental principles outlined in the previous articles. Now let's discuss the details of the WWW server environment and the numerous technologies and standards that support the development environment, which together make up our open systems development environment model. It is

important to keep in mind, however, that the most important principle is still to use proven open technology standards.

A WWW server provides client browser access to the numerous systems and associated data. This includes legacy system data, that those systems contain based on the instructions generated by the development environment that supports the server. It is important to note when discussing WWW servers that this can involve hundreds or even thousands of interconnected servers, platforms, gateways and databases that may be located anywhere the Internet can reach. Later I will examine how this will also be true of the development environments that we will have access to in the near future.

Terminal emulation gateways support TN3270 access to CICS, DB2 and IMS information systems found on mainframe computers. TN5250 terminal emulation supports access to AS/400 midrange computers as well as the terminal emulation for other midrange computer vendors. VT100 terminal emulation supports access to UNIX and other PC type systems software functions.

CGI gateways support access to CICS, DB2 and IMS information systems and many others found on all platform types. CGI gateways also support many of the middleware functions required for several of the application development tools being used.

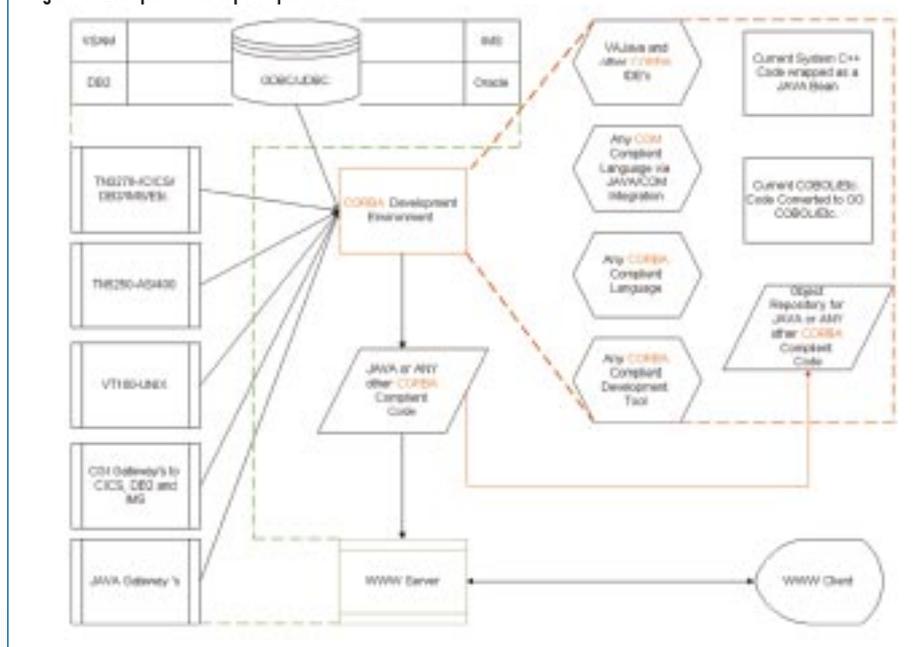
Java gateways provide the same functionality as the CGI gateways with the benefit of Java's ability to offload the majority of CGI's cumbersome server side processing requirements to the client machine. The advantages of Java make it likely that CGI technology will become obsolete if all Java security issues have been eliminated.

## Our open development environment model consists of two major areas of activity: the development environment and controlling software configuration software (SCM) software.

Database bridges using ODBC and JDBC technology provide direct access to database environments such as Oracle, DB2, IMS and VSAM. The advantages of JDBC make it likely that ODBC technology will become obsolete if all Java security issues have been eliminated.

These WWW servers are the supporting infrastructure that allows us access to our important legacy system assets.

Figure 1: Example of Concept Requirements



### THE FUTURE OF APPLICATION DEVELOPMENT

The most significant advantage that our open systems development environment model provides is the ability to more easily migrate into a fully functional object-oriented environment such as CORBA. As the foundation for this example, I will be using IBM's enterprise solution (e-business) for building distributed applications, however, all aspects discussed apply to any vendor's CORBA implementation.

The central nervous system of this model is the SCM program that controls the development environment. In a CORBA world this SCM is IBM's Component Broker for OS/390, which will help us develop and deploy distributed component-based CORBA applications. Component Broker is an enterprise solution for distributed object computing, providing a scalable, manageable runtime for developing and deploying distributed component-based solutions.

Component Broker allows networked applications to use, coordinate, store, translate and sort information from disparate existing or new applications and back-end systems. IBM's Component Broker or equivalent software, which manages the CORBA environment, will ultimately manage the global development environment. This global development environment encompasses application objects stored on other servers and in repositories that will be shared by companies anywhere in the world that the Internet reaches.

We have just seen how the WWW server environment encompasses the global environment, which is an integral part of the business model. We are now exploring how the development environment itself will become a global development environment. The complexity of application development in relation to application objects that will have to be manipulated and the eventual global location of these objects will require CORBA development tools that provide capabilities similar to VRML (Virtual Reality Modeling Language) technology.

VRML is a "bleeding edge" technology and extremely complex language because of its necessity to maintain constant and direct involvement in every object of its nodes. VRML can only be efficiently programmed using a visual modeling tool. The only reason to learn the language itself is to perform minor tweaking and performance modifications to enhance its final result. All of the development and modeling concepts inherent in VRML technology can be directly applied to future application development and the requirements of future CORBA development tools.

As I mentioned in last month's article, to prepare for using VRML type application development tools in the future, GUI and WYSIWIG development tools will be mandatory in our open systems development environment model and are vital to the future success of application development in the 21st century.

VRML-type CORBA development tools are a few years away, but now is the time to

start thinking about migration into a fully functional object-oriented environment such as CORBA. The first step is the creation of an open development environment that supports GUI and WYSIWIG development tools. I predict that a company's ability to utilize this evolving technology will be directly proportional to its future success.

### SUMMARY

While the goal of this series is to demonstrate how we are creating an open systems development environment model, there is also an underlying multifaceted objective to the goal. I stated in the first article of this series that it is important to understand that I am not proposing implementing object technology as the short-term solution to development requirements. However, this final article should make you acutely aware that your long-term goal should be to start implementing object technology. The "bleeding edge" application development tools that will be available to those companies using object technology will play an important role in the company's future success.

This final article of the series is by no means the end of the articles that will be created based on my continuing research and development efforts. Future articles will continue to build upon these important goals that are destined to transform our profession and application development as we know it. Your comments and suggestions for future articles will help provide new and stimulating dialog on this important subject.

Reader Brian Davis has brought to my attention that the information provided here and any updates need to be made available for companies making these types of decisions. In response, I have created a web page to address this need. The web page contains the latest up-to-the-minute work that I am doing on legacy systems enablement as well as a mailing list to notify subscribers of important updates. The web page can be found at [www.vipond.ac/](http://www.vipond.ac/) by clicking on the “Legacy” animation. 

---

*NaSPA member Richard B. ViPond is a senior consultant for Cyber Network Services, Inc. He can be reached at [rich@vipond.ac](mailto:rich@vipond.ac).*

*A special thanks to NaSPA members and technical editors, Dwight S. Miller and Stephen J. Pryor for their help with this series.*

©1998 Technical Enterprises, Inc. For reprints of this document contact [sales@nasma.net](mailto:sales@nasma.net).

