Leveraging The Power Of Abstraction In Enterprise Level Application Deployment

Sanjay Chandru
Joel Duquene

Background

The need to develop reliable applications efficiently has never been more pressing than it is now. So is the need for tools that provide integration between the traditional legacy applications and newer technologies. Enterprise developers are faced with greater challenges than ever in trying to strike a balance between the old and the new for achieving increased productivity. They need to be able to maintain existing systems, while at the same time acquiring new skills. The existing legacy applications and the information that they access are the heart and soul of the world's largest companies and must be maintained and protected. Companies are faced with the need to reallocate resources already stretched thin without eliminating existing systems which have been put in place at considerable cost in resources and money. Enterprise developers have an in-depth knowledge of existing applications and new employees need to understand them if the integration process is to succeed. The drive to the Web and e-business is shortening an already strained development cycle. Having the right tools to quickly develop and deploy stand-alone and distributed enterprise applications is essential.

Web development: Because of its portability and breadth of support across many platforms, Java™ has become one of the most popular programming languages for Web development. Java, combined with XML, is ideal for deploying applications across the enterprise. Many Java-based tools are available to make it easier to develop Web applications. With the Java™ 2 Enterprise Edition (J2EE™) features such as Enterprise JavaBeans™ (EJB), JavaServer Pages™ (JSP), and Java Message Service (JMS), it is even easier to deploy Java applications across the enterprise.

Client/Server development: With client/server, companies are able to put Terminal User Interface (TUI) and Graphical User Interface (GUI) front-ends to their enterprise applications. The host (server) application traditionally provides access to the back-end database and the TUI or GUI client does the presentation. Prior to the emergence of the Web browser as a user interface, this has been one of the most favored ways to provide a front end to host applications.

Traditional host development and web technologies, can they co-exist?

Third generation languages such as COBOL and PL/I have been the implementation vehicles of choice for enterprise developers. With so much investment in existing COBOL applications, it becomes prohibitively expensive to throw them away in favor of applications deployed with newer languages and tools. At the same time, it is becoming increasingly difficult to find people with skills that relate to both mainframe and workstation environments. Thus, it becomes important to utilize tools that can allow the integration of existing and emerging technologies (Java, Web, JSP, EJB, etc.). This integration with the

existing applications can be made easier if they were designed in such a way that there is a clear separation between the presentation and business logic.

Tools that can allow existing TUI applications to communicate with Java-based and web-based tooling can dramatically improve the productivity of an enterprise. This means that developers could spend less time converting existing applications to leverage the power of the web, and invest time in other more useful endeavors.

A very important aspect of tooling an enterprise lies in the ability of the tools to support easy portability and scalability of applications. It is important for development environments to allow an application that is written for the workstation to be ported to a mainframe environment or vice versa if a need arises in the future.

There are currently very few application development tools that allow that to be done with ease. Furthermore, there are very few code generation tools available in the market that allow an enterprise developer to write and integrate mainframe applications with workstation and web-based applications without jumping through hoops. This allows a true 3-tier server driven architecture that permits the separation of presentation, business logic, and processing such as number crunching and database access.

A common problem faced by an enterprise developer is the integration of the presentation layer and enterprise back end processing layer. How can issues such as state management, host management and data connectivity be handled without investing heavily in new training and allocation of resources that are not available?

IBM® currently offers several technologies that allow the enterprise to leverage and integrate the latest technologies with existing frameworks, without compromising on flexibility and ease of use.

We will be looking at how VisualAge Generator can bring new and existing technology together to provide a solid application development platform for the enterprise.

Accessing host applications from newer technologies

One of the biggest challenges facing the enterprise developers is how to make the host applications accessible to newer technologies. There are many options available.

Web Tooling

Java (applet, servlet, JSP), Perl, CGI, and ASP are some of the popular choices for web development languages.

There are many great tools available for Web development and deployment.

- **IBM VisualAge® for Java** for developing Java applications (Applets, Servlets, JavaBeans™, Enterprise JavaBeans, etc.) that are accessible from the Web. With the Integrated WebSphere Test Environment, developers can quickly find out how their code integrates with the Web prior to being deployed to WebSphere
- **WebSphere Studio** for building Web pages, developing Java Server Pages (JSPs) and integrating them with Java Servlets and JavaBeans. Web Pages can then be deployed to VisualAge for Java for testing and to WebSphere for production
- **WebSphere Application Server Advanced and Enterprise Edition** for Web and application deployment. It provides full support for the Java 2 Enterprise Edition, CORBA and other Web-related technologies. Applications developed with VisualAge for Java and WebSphere Studio can be easily deployed to WebSphere Advanced Server

These tools provide versatile and easy development and deployment environments for a developer. For those who desire to do Java development, VisualAge for Java is the right tool. For building and deploying Web application, WebSphere is the tool of choice. WebSphere is available for OS/390®, which enables host applications to leverage the power of the web. Since it is available for a variety of platforms, standard development practices can be adopted within an enterprise using these product. However, they require traditional enterprise developers to learn new skills in order to be able to provide Web access to their legacy code. On the other hand, the Java developers need to understand the legacy applications in order to be able to integrate traditional legacy applications with newer technologies. Although these products are versatile and powerful, training is required for effective integration. But there exists an option that allows enterprise developers to leverage the power of WebSphere Advanced Server Enterprise Edition and VisualAge for Java. A tool that also provides a means of integrating tier-1 and tier-3 without training and reallocation of resources. This application development tool is VisualAge Generator V4.5

Choosing the right tool; VisualAge Generator as the Tool of Choice for Enterprise Application Development

The goal of the enterprise is to be able to quickly extend the reach of their legacy applications without too much expense.
Code generation software has been around in different forms for several years now. The basic purpose of code generators is to reduce complexity in development and deployment of applications in an enterprise. There are several software vendors who provide enterprise level code generation software. We will proceed to explore and highlight features and advantages VisualAge Generator provides over these software packages. This paper attempts to give a high level overview and does not attempt to get too detailed about functionality. An attempt has been made to provide enough information for customers to be well informed without being overloaded with technical specifications.

There are three main areas to be examined when determining the versatility of rapid application development tools

- **Rapid Application Development**
  - Good visual development interface for shorter development times
  - Availability of templates and business objects to ease development process
  - Rapid application development script that abstracts over traditional languages and performs functions that ordinarily need to be done manually by the enterprise developer. A good example would be transaction control, which can be invoked with a single function call in the VisualAge Generator script, without the user having to worry about how commits and rollbacks are performed in different programming environments
  - Proper partitioning of clients, data and processes
  - Incremental development and testing
  - Strong team programming environment, facilitated by repository management and change control
- **Generation**
  - Support for varied client implementations
  - Code generation in widely used languages and many popular platforms
  - Ability to compile and run the code without having to manually perform these functions
  - An end to end solution for varied applications
  - Code generation for access to many different data sources, such as DB2®, VSAM, Oracle, DL/I.
- **Execution**
  - Allow n-tier implementations for efficient utilization of resources. This also permits separation of business logic and presentation
  - Middleware transparency for saving in training costs
  - Portability and scalability to enable easy redeployment of existing applications in new environments
  - Runtime support for access to many different data sources, such as DB2, VSAM, Oracle, DL/I.

There are different types of rapid application development tools in the market today that are each useful for specific development environments. A few important questions need to be pondered before purchasing application development (AD) tools.

1. Can an enterprise afford to utilize tools that are not flexible enough or comprehensive enough to not allow for future change?
2. What is the impact of hiring new talent to keep up with all the latest greatest technology without sacrificing existing frameworks and talent?
3. Are there any tools that can allow easy development and deployment to most of the major platforms and environments with low to no overhead?
4. Is it important for new applications to be able to integrate and interact with existing applications in various development environments?
5. Is there a rapid application development tool out there that can give the enterprise everything that is needed without sacrificing all the investment in existing and old technologies?

It is becoming increasingly evident with all the new technology being introduced everyday, it is costly and unfeasible to upgrade every time. Another major factor being cost of training employees to utilize all this new technology. Managers must therefore weigh the pros and cons of adopting new technology versus keeping the old. This often turns out to be a crap shoot. To prevent a costly mistake organizations must be prepared to migrate existing applications to new platforms and programming environments that might allow the introduction of newer technologies. Therefore, it becomes imperative to use an application development tool that allows existing applications to be redeployed with minimum effort. VisualAge Generator achieves this and more with scalable application development and extensive platform support.

Platforms Supported by VisalAge Generator V.4.5
OS/390 (CICS®, IMS®, Batch, TSO)
VSE® (CICS, Batch), VM
OS/2®, OS/2 CICS
AIX®, AIX CICS
Sun Solaris ™, Sun Solaris CICS
HP-UX®
AS/400®
Windows NT®, Windows NT CICS
Windows® 2000 and Windows 2000 CICS
Windows® 95, Windows® 98

There are several features available in VisualAge Generator that sets it apart from the competition. VisualAge Generator V4.5 is available in two flavors, VisualAge Generator for Smalltalk, and VisualAge Generator for Java. Some of the more important features allow for significant improvement in productivity and efficiency.

- Repository management
- EZE words (Special function words) for quick and easy access to various functions including transaction control
- Reusable parts, which facilitates lower development times
- Interactive test facility (ITF), which allows code to be tested before compilation. Business logic can be debugged before code deployment. This results in cleaner and significantly quicker code development
- Dynamic Program Partitioning, for streamlining client-server application architecture
- MQSeries® support built into the product
- Web transactions to allow true 3-tier web application deployment
- Host development covering a variety of host platforms
- End to end development, which includes generation of code in either C++ and Java (for workstation platforms) or COBOL (for OS/390, VSE, VM and AS/400)
- Generated code is also compiled and ready to execute. This takes away the need to understand an often cumbersome compilation and link process for development environments

Interactive test facility is a very versatile tool within VisualAge Generator that allows testing of business logic before compilation and deployment of code. Code being tested behaves as it would if it were actually deployed in one of the environments supported by VisualAge Generator under most circumstances. A test monitor window allows users to follow the exact rapid application development code being debugged. Different views include execution stack monitor, watch point monitor and statement monitor, which allows users to monitor the statement being executed. A trace log displays specific user requested information and data. Several trace logs can be open during debug, each displaying different information. Data values can be modified during execution time to alter the behavior of applications. Breakpoints can be set and the rapid application development script can be changed during debug time, which allows dynamic logic correction. ITF even allows database access, client server testing, and external non VisualAge Generator program calls during the test cycle. File I/O is done with relative ease from ITF. DB2 and VSAM file access, DL/I calls, SQL calls, and MQSeries queue access are allowed from within ITF. During ITF testing, VSAM and DB2 access can be local or remote in a Windows NT environment. DL/I calls are remote. There are tremendous savings in time and developer efficiency is increased several fold due to the versatility of ITF.

Dynamic Program Partitioning is a feature within ITF that helps developers determine how best to partition distributed applications for optimum performance. Applications can be configured dynamically during the test phase within ITF. Dynamic Program Partitioning highlights changes required in structure and distribution of applications for better performance. Each program is a floating icon, with programs that communicate extensively and pass large amounts of data clustering together. To take advantage of this feature, application logic should be broken between several programs. The programs will “float” to the target machine where most of their interactions take place. This usually results in optimum partitioning.

MQSeries access from VisualAge Generator V4.5 is extremely simple. MQSeries queue access is achieved through VisualAge Generator I/O verbs. A lot of work is done under the covers and frees up the developer from having to learn MQSeries API implementation. All the developer needs to understand is the need for MQSeries within the application architecture.

Some of the advantages offered by VisualAge Generator V4.5 support for MQSeries are

- A higher level of abstraction over MQSeries API calls
- ADD and SCAN and CLOSE I/O options instead of coding MQSeries API calls
- MQSeries functionality can be exploited without any complex code
- Simplified queue access when compared to a 3GL language like C or C++
- Queues are treated as files
- File level support including:
  - Automatic connection to queue manager
  - Automatic opening of queues
  - Automatic closing and disconnecting
  - Automatic return code checking
  - Automatic data conversion
  - Optional termination on hard errors

- Optional access to MQSeries control blocks
- Transaction control using VisualAge Generator special function words

With VisualAge Generator, the developer generates applications that access many different data sources, such as VSAM, DL/I, DB2, Oracle. They can use ITF to test those applications from their desktop before they get deployed to the target system. The data source can be accessed remotely for DL/I and either locally or remotely in the case of VSAM and DB2. This is a great boost in productivity.

VisualAge Generator offers the user the advantage of not having to learn the changes needed to deploy MQSeries applications for different platforms. A complex workstation application can be generated and deployed on OS/390 or VSE CICS without any changes to the application architecture. This could mean a dramatic decrease in development and deployment times. Also the need for learning C, C++ or Java to write MQSeries applications is eliminated. MQSeries API implementation and parameters needed to access various functions are hidden from the user by an easy to understand rapid application development script and interface.

Web Transactions: Minimum effort, Maximum gain

Web systems are evolving to a 3-tier server-driven architecture (e.g. Servlets, JSPs). But there are challenges to be overcome before an enterprise can make a successful transition to web based applications. Web Application Servers (e.g. WebSphere) provide tier-2 runtime and middleware connectivity & integration to tier-3. But in a conventional implementation users have to worry about state management, and other tier-2 issues. Web transactions in VisualAge Generator V4.5 allows users to develop web enabled applications with a true 3-tier implementation which allows separation of business logic and presentation (web based customizable user interface). Back end systems perform the database processing, tier-2 manages state and data connectivity. Tier-2 implementation is managed in its entirety by VisualAge Generator, thereby saving time and money for the enterprise. Results of any processing are presented to customers through a generated customizable web user interface.

Web transactions,

- Provide simple rapid application development single threaded programming model
- Generate JavaBeans & JSPs and provide runtime support on tier-2
- Automatically connect to tier-3
- Allow web developers to use standard components: JSPs and JavaBeans
- Hide tier-2 programming complexity, yet exploits capabilities
- Enable separation of concerns:
  - Shields Enterprise Developer from Web UI design/development
  - Shields Web Developer from business function and back end processing concerns
- Utilize a programming model which is familiar to the enterprise developer
- Can be adopted with very little training costs. A huge saving in time and effort

Summary

As technology evolves, so should the enterprise. Adopting new technology can often be a difficult experience, if the right tools are not available to developers. Often, managers shy away from change precisely for this reason. VisualAge Generator V4.5 makes the decision to integrate new technologies easier, by providing a variety of useful features that result in big cost and resource savings. In the long run, application development with VisualAge Generator V4.5 will pay off dividends in terms of easy and quick adoption of new technologies, as well as lower training costs.

References

(1) IBM VisualAge Generator User's Guide Version 4.5, 2000 SH23-0268-01
(2) Enterprise JavaBeans Development Using VisualAge for Java, IBM Redbooks 1999

For more information about VisualAge Generator

www.software.ibm.com/ad/visgen

Special Notices

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)®
IBM ®
AIX
CICS
IMS
VSE
DB2
OS/2
OS/390
AS/400
MQSeries
MVS/ESA
VisualAge
WebSphere

The following terms are trademarks of other companies:

Sun, Solaris, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Microsoft®, Microsoft Windows®, Microsoft Windows NT®, Microsoft Windows 2000® and the Windows® logo are trademarks of Microsoft Corporation in the United States and/or other countries.
HP, HP-UX are trademarks and registered trademarks of Hewlett Packard, Inc. In the United Stated and/or other countries.
UNIX® is a registered trademark in the United States and other countries licensed exclusively through The Open Group.
Other company, product, and service names may be trademarks or service marks of others.

Disclaimer

IBM has made reasonable efforts to ensure the accuracy of the information contained in this article. However, this article is presented "as is" and IBM makes no warranties of any kind with respect to the contents hereof, the products listed herein, or the completeness or accuracy of this article. Customer experiences may be different from those described here. IBM does not warrant any non-IBM programs or products, which are described in this article. This article is for information only, and you should contact the stated company with your questions.