VisualAge® Generator – Past, Present & Future

by Steve Choquette, VisualAge Generator Product Manager

Past

Twenty-one years ago in computing, there was no TCP/IP (1982), no Windows® (1985), and no CD-ROMs (1985). IBM® had just introduced its first PC (the IBM 5150); for $2880 you could get 64K RAM, an Intel 8086 processor, and the IBM/Microsoft PC-DOS operating system. Bill Gates had uttered his famous "640K ought to be enough for anybody" statement. Mainframe programmers, which comprised the vast majority of the computing community, used COBOL®, PL/1, C, or Pascal to write the programs that would help the world's largest companies be successful. The VAX-11/780 was a popular mini-computer. The Cray 1 was the first commercially available supercomputer. The S/370-Extended Architecture was announced, providing an unprecedented 2 gigabytes of addressing for real and virtual memory. The Internet had 256 computers on it.

In 1981, IBM introduced an application development tool named **Cross System Product (CSP)** (http://www-4.ibm.com/software/ad/visgen/csp/). Cross System Product was a mainframe application development tool intended for use by data processing professionals to increase their application development productivity. To improve programmer productivity, the product used a COBOL-like scripting language called 4GL. 4GL shielded the programmer from the programming complexity unique to each target operating platform and data store. To go to a different target platform, a user simply had to regenerate; no source code changes were required. The 4GL also provided the means to quickly validate and unit test your source code.

Cross System Product evolved over the following 11 years. Initially designed to support mainframe file systems such as Virtual Storage Access Method (VSAM), the product was gradually extended to include support for emerging database technologies. In the mid-1980s, Cross System Product enjoyed tremendous success in the marketplace as the rapid application development tool for the rapidly emerging relational databases. Cross System Product V4.1, introduced in 1992, switched from an interpretive environment to a compiled COBOL environment, significantly improving the performance of applications in the MVS® execution environment. Along with the COBOL code generation, V4.1 also introduced a workstation development environment through the CSP/2AD feature; this allowed for improved application development productivity by preventing errors, reducing keystrokes and exploiting a multi-tasked windowed environment. This feature allowed the creation of new applications and the editing of old ones using a mouse and a graphical interface. By the early 1990s, Cross System Product was in use at over 5500 installations, including some of the world's largest banks, retail, insurance, and manufacturing enterprises.

In the early 1990s, a large number of IBM customers were interested in taking advantage of the workstation for application development. Additional requirements included the ability to develop client/server applications (particularly the addition of Graphical User Interfaces (GUIs) to applications), the ability to access data from non-IBM vendors' data stores, and the ability to execute application in operating environments beyond the mainframe. IBM delivered its response to these requirements in the summer of 1994 with a product called **VisualGen**®. VisualGen provided the means to remotely test a program on a workstation using real data, and then move it to the appropriate target platform for production verification.

Although this new product could have been called Cross System Product V5.0, IBM chose to change the product name because this version was so much more powerful than Cross System Product. VisualGen was built around emerging object-oriented technology to support GUI construction, but at its heart it preserved the same 4GL that was used by its predecessor. That meant that ALL Cross System Product users could migrate their skills and applications to this new product. The new product was not identical to Cross System Product (e.g. new repository structure, workstation-based), but migration tooling was provided to ease the migration to VisualGen.

The product was renamed a second time in 1996 to **VisualAge**® **Generator** to position it properly in IBM's family of workstation Application Development products called VisualAge.

While Cross System Product and VisualAge Generator have evolved significantly over the past 20 years to accommodate a multitude of architectures (TUI, GUI, web transactions), data stores (file managers, native RDBMS, ODBC, JDBC), transactional system (CICS®, IMS®), and technologies (e.g. MQSeries®), the core problem to be solved remains the same:

> **Provide an integrated tools environment for the rapid development of scalable, robust, mission-critical applications using traditional enterprise skills capable of running under a variety of environments and topologies.**

VisualAge Generator can solve this problem

- through a platform-neutral development language, 4GL, which is transformed into the language best suited for the selected target platform (COBOL, C++, Java™)
- through an optional high productivity and customizable facility that allows automatic generation of database-oriented applications, called VisualAge Generator Templates.
- through a workstation-based test facility that permits stepping through the source code using real data before the application is deployed into the target environment
- through 21 different target platforms from the largest mainframe running zOS® to the smallest workstation running the most-recently announced Windows XP® operating system

If you are a Cross System Product customer, you should be aware that V4.1 went out of support on December 31, 2001 after 20 years of service. So while your applications will continue to run, you can no longer call IBM for Q&A and defect support. If you are currently migrating from Cross System Product to VisualAge Generator, you can purchase up extended V4.1 support through December 2002. Contact our VisualAge Generator Support Manager (vagen@us.ibm.com), if you are interested. The current release of VisualAge Generator, Version 4.5, includes tooling that helps migrate from Cross System Product V3.2.2 or higher to VisualAge Generator. A strong community of highly skilled business VisualAge Generator business partners can also provide help with education, consulting, migration services, and even migration tools from competitive 4GL solutions.

Present

VisualAge Generator V4.0

In September 1999, IBM introduced VisualAge Generator V4.0. This release provided the web transaction capability to produce applications that are accessible through the Internet. Web applications, often-called "n-tiered" applications consist of a client tier (typically browsers and thin clients), a middle tier (web and application servers), and a back-end tier of legacy applications and databases.

Browsers generally remove the requirement to distribute code to each user of an application. This greatly simplifies the task of managing the deployment of applications as enterprises have hundreds to thousands of internal users and thousands to millions of external users. As new client hardware platforms such as hand-held devices and data-enabled phones emerge, efficient application designs can avoid pushing large amounts of code to the client.

The middle tier(s) play a critical role in a Web application. It is essentially the glue that holds everything together. For instance, the middle tier is responsible for managing communication with both the client tier and the back-end tier. The middle tier often leverages JavaServer Pages (JSPs™) to manage both input from the client and the presentation of information back to the client. The middle tier manages communication with the back-end server.   For VisualAge Generator web transactions, this middle tier code - JSPs, JavaBeans™, Enterprise JavaBeans (EJBs™), and servlets - typically runs under the WebSphere® Application Server.

The back-end tier is where the bulk of an enterprise's current information technology investment resides. It often includes transaction processing applications and databases that are critical to the business' day-to-day operations. These applications and databases have often been developed over very long periods of time. This is the core functionality that enterprises want to leverage as they build web applications. Doing so can save them significant time, energy and money.

With VisualAge V4.0, a user can generate a turnkey (end-to-end) web solution without knowing Java. VisualAge Generator produces the JSPs, which interact with the user via a browser, the middle tier Java code running under an application server, and the back-end server code (often a COBOL application) that interacts with a database.

Consultants estimate that it would take 1-2 years to retrain a COBOL programmer in Java so they could write a web application. With VisualAge Generator, today's COBOL programmer can quickly write applications for the web. The Commonwealth of Kentucky found this out when they turned to VisualAge Generator to produce a web application that tapped into an IMS® database to manage the Commonwealth's fleet of trucks and cars (http://www-4.ibm.com/software/ad/visgen/library/). Many other customers are discovering the power of VisualAge Generator web transactions to conduct online banking, produce employee portals, and manage department stores.

VisualAge Generator V4.0 also introduced integration with the VisualAge Java product. This meant that customers could have an Integrated Development Environment based on either VisualAge Smalltalk or VisualAge Java.

VisualAge Generator V4.5

The latest release of VisualAge Generator, V4.5, shipped in September 2000 and included the following enhancements:

- Integrated support for MQ Series®
- Java generation for back-end servers on Windows NT® and Windows 2000®
- Closer integration with WebSphere Studio
- AS/400® enhancements for 2-tier and mid-tier support, Remote OS/400® File Access
- Windows NT and OS/2® Installation improvements
- Support for SCO Unix OpenServer platform as a back-end server for TUI applications
- Support for Solaris as mid-tier and back-end server for Java and GUI clients
- Remote VSAM Access from ITF
- Remote DL/1 access from ITF
- VisualAge Generator Templates
- BIDI Conversion Table support for Java clients

Since VisualAge Generator V4.5 shipped, numerous enhancements have been added to the product via fixpacks and host Program Temporary Fixes (PTFs). These enhancements continue to focus on the concerns that motivated customers to purchase Cross System Product and VisualAge Generator in the first place – platform neutrality, high productivity for enterprise programmers, and scalability across a number of target environments. Listed below are some of the major enhancements from Fixpacks 1, 2, and 3:

- Performance improvements to host and C++ run-times
- Improvements for IMS applications

- Ability to run VisualAge Generator on VA Java 3.5.3 and 4.0
- Java generation for OS/390®, OS/400, and variety of Linux platforms
- Windows XP® can be used as a development, mid-tier, and back-end server platform
- Ability to operate ITF in EBCDIC mode
- Enhancements for Web transaction applications
- Ability for VisualAge Generator web transactions to run under WebSphere ® Application Server (WAS) V4.0

VisualAge Generator V4.5.4

VisualAge Generator V4.5 can be installed as a feature on top of VA Java 3.5 and VA Smalltalk 5.0. Both of these releases, as well as various other VisualAge Generator pre-requisite and co-requisite products, will be going out of support in 2002. In addition, V4.5 has an announced end-of support date of December 31, 2002. Later this year, this date will be extended to June 30, 2003. Support for the OS/2, CICS OS/2, Windows NT, SCO OpenServer, and HP-UX platforms will not be provided during the extension period (January 2002 to June 2003). Other platforms shipped with V4.5 will be supported during this extension period.

To further the life of VisualAge Generator, the lab team is investigating the shipment of a VisualAge Generator V4.5.4 release in the second half of 2002. This release, which is primarily a repackaging of VisualAge Generator V4.5, would be a means to move the product on top of pre-requisite and co-requisite versions that will be supported for many more years. Support for OS/2, CICS OS/2, Windows NT, SCO OpenServer, and HP-UX platforms will not be provided in the new V4.5.4 release.

VisualAge Generator Host Services 1.2 also has an announced end-of-support date of December 31, 2002. Later this year, this date will be extended through 2004.

The year 2001 marked the end of support for VisualAge Generator releases 2.2, 3.0, and 3.1. VisualAge Generator 4.0 went out of support March 31, 2002. This means you cannot call IBM for Q&A or to report defects for these releases. VisualAge Generator V4.5 includes tooling to help you migrate from these back-level releases to VisualAge Generator V4.5. A strong community of highly skilled VisualAge Generator business partners can also provide help with education, consulting, and migration services.

VisualAge Enterprise Suite

In 2001, IBM also introduced a packaging option called VisualAge Enterprise Suite (VAES). VAES is simply a packaging offering of enterprise-type application development tools. See www-4.ibm.com/software/ad/vaes for details. VAES R4, introduced March 2002, contains the following products:

- VisualAge Generator Developer V4.5

- WebSphere Studio Application Developer – Integration Edition V4.01
- VisualAge COBOL for Windows NT V3
- VisualAge PL/1 V2.1
- Versata Studio Client 5.1

As your needs change from traditionally developed mainframe-based applications to those of an e-business, the IBM VisualAge Enterprise Suite V4.0 gives all of your developers tools to meet both of these business environments. Each licensed developer in your organization is entitled to use any of the products in the Suite. Different tooling may be used depending upon the skill of your developers and the needs of a particular project.

Future

As the industry has changed over the years, so has VisualAge Generator. Today's applications are being built to run on the web. The terminology being used today was not invented 20 years ago – STRUTS, XML, web services, SOAP, UDDI, Java, components, pervasive, EJBs, application servers, transcoders, portals, wireless, J2EE™. The problem to be solved remains the same:

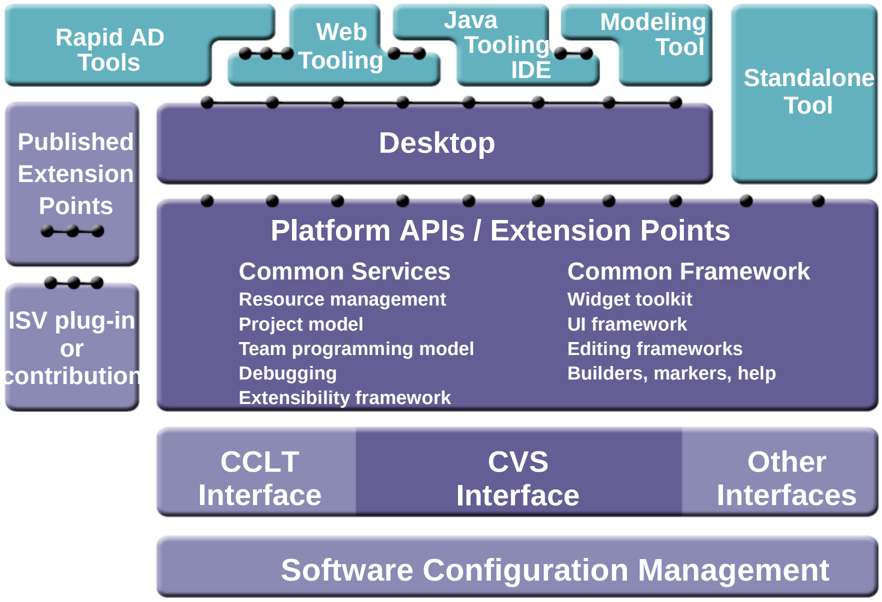
> **Provide an integrated tools environment for the rapid development of scalable, robust, mission-critical applications using traditional enterprise skills capable of running under a variety of environments and topologies.**

WebSphere Studio Workbench

If you were at a VisualAge Generator or WebSphere conference, or have read recent announcements from IBM, by now you have heard about a project code-named the "Eclipse". "Eclipse" is software IBM developed that would provide a foundation on which IBM and ISVs would build future application development tools. IBM donated most of the workbench code to a new open community (see www.eclipse.org). The old code-name of "Eclipse" now refers to the open source of the IBM workbench code.

IBM used the Eclipse base to produce the **WebSphere Studio Workbench** (see http://www-4.ibm.com/software/ad/workbench/ and the graphic below). The WebSphere Studio Workbench is a comprehensive technology that offers a complete tool construction framework to provide:

- Integration with other workbench-based products
- Code, project and resource management services
- Test and debug facilities
- UI and text editor frameworks
- Widget toolkit
- Help systems



The Workbench provides a Version & Configuration Management mechanism that allows Software Configuration Management (SCM) manufacturers to write an adaptor to connect their company's repository to the WebSphere Studio Workbench. This means that it is now possible to use the same enterprise repository with a wide variety of development artifacts (e.g. JCL, JSPs, PL/1 or COBOL programs, 4GL source, EJBs, design specifications). For VisualAge Generator customers, moving to the WebSphere Studio Workbench means migration from Envy (a proprietary repository) to the repository of your choice. In fact, you may migrate to the same repository you already use today with VisualAge Generator to store your generated COBOL code.

WebSphere Studio Tools

On November 5, 2001, IBM announced a new family of WebSphere application development tools that deliver the industry's broadest support for J2EE, Web services, XML, HTML, rich media, site design, voice, wireless and embedded devices. See http://www-3.ibm.com/software/info1/websphere/news/ibmnews/pr011105.jsp for the details regarding the announcement.

IBM's new WebSphere Studio tools are the first commercially available tools built on the WebSphere Studio Workbench. Developers working on WebSphere Studio and other Workbench-based tools use a common, easy-to-use interface that provides a consistent "look and feel," regardless of vendor, which can significantly reduce training requirements.

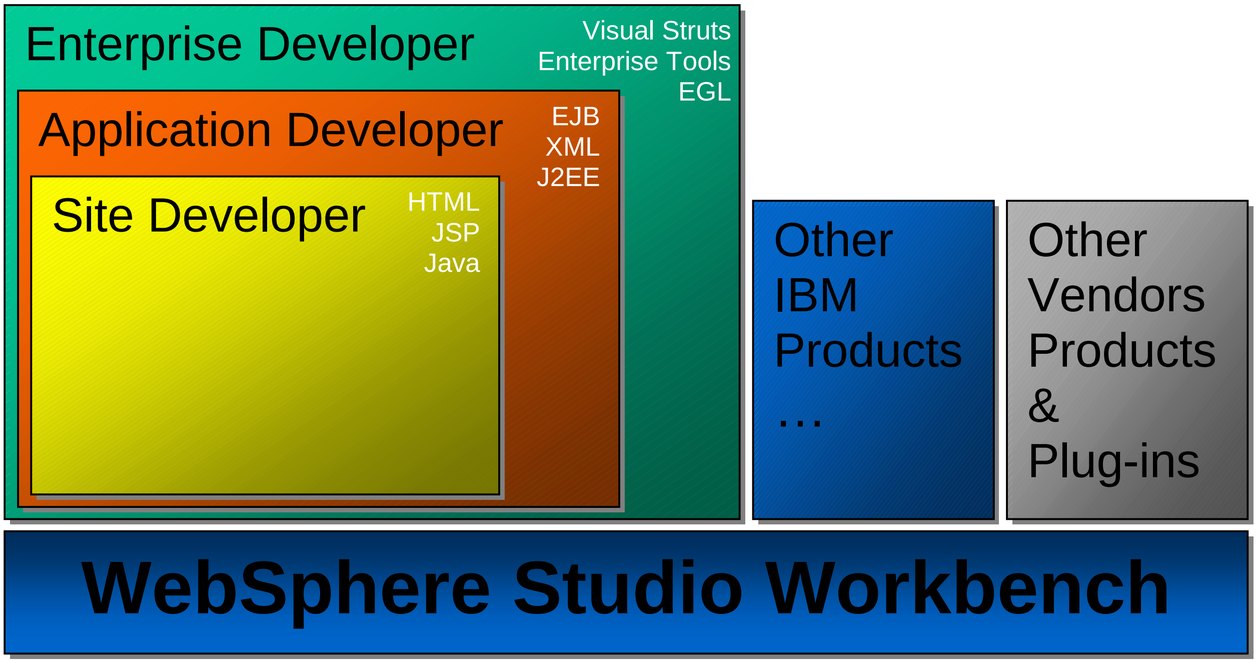
At the time of the writing of this paper, over 1200 third-party software vendors, including industry leaders such as Rational Software, TogetherSoft, Holosofx, Serena, Merant, Sitraka, Mercury Interactive have announced their intention to integrate their tools into the IBM WebSphere Studio Workbench.

This unified interface and easy integration encourages team-oriented programming and developer collaboration among people with different skills, such as programmers, web content developers, business analysts, database administrators, wireless and voice application developers, and graphic artists. It can also enable developers to customize their environment and mix-and-match tools of their choice.

In addition, WebSphere Studio tools enable developers to create applications and test them on middleware, all within the same environment. WebSphere Studio Tools provide a common development environment across Windows and Linux, so Linux developers can create enterprise-ready applications directly on top of Linux, without having to port them from Windows. This saves time and creates higher-quality applications.

IBM's new tools, all based on the WebSphere Studio Workbench, include:

- **WebSphere Studio Homepage Builder** — This is targeted to individual Web site developer, and delivers an easy-to-use set of tools for developing sites with rich media content including animation, audio and video. This product is currently available.
- **WebSphere Studio Site Developer** — provides tooling for professional web site developers, including sophisticated web page authoring facilities, site web site management, Java programming, XML and web services development facilities, application deployment, and more. See http://www-4.ibm.com/software/ad/adstudio/about/sitedeveloper.html for details. WebSphere Studio Site Developer (WSSD) is the follow-on product to WebSphere Studio.
- **WebSphere Studio Application Developer** (WSAD) — the industry's first tool to integrate a complete, powerful J2EE development facility and web development tools into a single environment. It is targeted to Java and J2EE developers for development of applications that require integrated EJB and JSP support. See www.ibm.com/software/ad/studioappdev for details. This product is currently available. WebSphere Studio Application Developer, as the follow-on product for VisualAge Java, includes full WebSphere Studio Site Developer functionality.
- **WebSphere Studio Application Developer for Linux** — for integrating Java and Web development tools on top of Linux. A full production version of this product will be available in early 2002.
- **WebSphere Studio Enterprise Developer** (WSED) — This is the WebSphere Studio family flagship product, targeted to the widest range of enterprise developers, and delivering visual composition of end-to-end enterprise applications that can run on WebSphere Application Server and integrate with traditional back-end transactional systems such as CICS and IMS. WebSphere Studio Enterprise Developer is the follow-on product for VisualAge Enterprise Suite. WebSphere Studio Enterprise Developer includes the functionality of WebSphere Studio Site Developer and WebSphere Studio Application Developer.



WebSphere Studio Enterprise Developer

At the heart of this new offering is familiar VisualAge Generator technology. But WebSphere Studio Enterprise Developer will be so much more than just the next release of VisualAge Generator or VisualAge Enterprise Suite.

WebSphere Studio Enterprise Developer addresses the following problem areas when building large-scale, dynamic applications deployable to WebSphere and traditional enterprise environments:

- Lack of integration of diverse skills sets across a multi-site enterprise
- Inability to reuse mission-critical legacy code when developing new applications
- Weak development processes and tooling for creation of web applications

WebSphere Studio Enterprise Developer solves these problems through the following approaches, which will be described in subsequent sections:

- Use of abstractions to leverage traditional enterprise environments (e.g. CICS)
- Use of componentization to encourage the rapid creation of well-structured web applications
- Use of visual assembly tooling to simplify the implementation of new and changing requirements

The initial release of WebSphere Studio Enterprise Developer, currently planned for 2H02, will focus on building web applications running in MVS environments under control of CICS. Subsequent releases of WebSphere Studio Enterprise Developer will provide more capabilities and target platforms that are familiar to VisualAge Generator and Cross System Product users.

Web applications produced by WebSphere Studio Enterprise Developer will take advantage of the latest Internet technologies - XML, Web services, STRUTS, for example. As with VisualAge Generator, this technology can remain invisible to the typical WebSphere Studio Enterprise Developer programmer. This is nothing new; a programmer using VisualAge Generator V4.5 web transaction support could produce an end-to-end Java solution without Java programming knowledge. This is the strength of the abstractions inherent in VisualAge Generator and WebSphere Studio Enterprise Developer.

Use of Abstractions to Leverage Enterprise Environments

The familiar 4GL scripting language of VisualAge Generator provides abstractions, which can allow a programmer to rapidly construct applications without knowing the intimate details of the particular technology. For example, with the MQSeries abstractions added in V4.5, a VisualAge Generator programmer can use the familiar ADD, SCAN, and CLOSE I/O verbs to add and remove messages from a MQSeries queue. These verbs replace hundreds of lines of detailed MQSeries API calls and return code checks with a very small number of 4GL statements to perform similar functions. The strength of a platform-neutral target language becomes very visible if the target platform (e.g. mainframe, workstation, AS/400), target language (COBOL, C++, or Java), or enterprise database (e.g. DB2, IMS) changes; the application would need to be regenerated and retested, but no coding changes would be required.

The familiar 4GL language is getting renamed to Enterprise Generation Language (EGL) with the WebSphere Studio Enterprise Developer. VisualAge Generator customers will recognize that EGL is extremely similar to 4GL, with the addition of functionality that has been requested by customers for many years via VisualAge Generator Supermarket Sweeps:

- Multi-level qualifiers
- Nested data structures
- CASE statement
- Multi-dimensional arrays

In addition, there will be syntax changes that make EGL consistent with the other languages used by WebSphere Studio Enterprise Developer programmers – Java, COBOL, JSPs, and Pl/1. Several examples of the syntax changes are: 1) EGL keywords can no longer be used as part names or variable names, 2) implied comments are no longer permitted, and 3) the TEST statement is now considered a special case of an IF statement. Migration tooling will be provided in subsequent releases to convert your existing 4GL programs to EGL.

Rapid Creation of Well-Structured Web Applications

A component can be any piece of code that provides a service to another aspect of an application. A component can be built in any language and may ultimately run in any environment. The use of components and component technologies can provide significant benefits in areas such as reuse, reliability, maintenance, scalability, and ultimate time to market of business applications. Component models allow teams to reuse proven and reliable runtimes and capabilities, instead of duplicating development effort.

Managing change is an important benefit of leveraging component architectures and models. If not managed well, modifications late in the development process or in subsequent releases can result in the degradation of a modern application into “spaghetti code”.

For web applications, change will be managed through a non-proprietary architecture called Model-View-Controller (MVC). MVC is a design for creating components that was put together to help manage change in application development. MVC separates the user interface (View) from the business logic and data (Model). The model and the view elements are the components driven by the Controller. This separation permits the development of adaptable, reusable, maintainable web applications. The work performed by the web page designer and the application programmer is better delineated and can, if desired, be performed by each team member independently.

STRUTS is a non-proprietary framework that will be used to implement the MVC architecture for separating the presentation and business logic. STRUTS is very similar to the existing web transaction approach used by VisualAge Generator programs today with JSPs for Views, Java beans to hold data for JSP usage, the specialized VAGen Gateway Servlet as the Controller, and actions analogous to web transactions.

See http://www-106.ibm.com/developerworks/library/j-struts/index.html for more information on STRUTS, http://ootips.org/mvc-pattern.html to learn more about the MVC framework. You can also go to the Apache Jakarta web site for information about STRUTS and MVC.

Use of Visual Assembly Tooling

With WebSphere Studio Enterprise Developer, the programmer has a choice in how to approach development of an application. They can use the traditional method of scribbling a design on a chalkboard and then mechanically transforming it into code within a development environment, or using a Visual Assembly Tool. The Visual Assembly Tool defines all applications in terms of their user interface (the View), and the business logic and data (the Model). This logical separation yields new applications that easily generate to the STRUTS – applications that are adaptable, reusable, and maintainable.

The Visual Assembly Tool facilitates a top-down approach to application development. In a single view, the designer can see an entire application. The designer can also "drill down" to see the individual components (for example, JSPs, EGL code, Java code) that make up the application. Different programmers can develop components at different times. When a specific component is "realized" (that is, when the component has been implemented), this change will be shown by the assembly tool.

The visual aspect of the tool also permits the rapid understanding of an existing application's structure and components - certainly one of the most significant challenges for a large Information Technology organization. The components of a STRUTS-based application will be displayed in a visual manner to facilitate application understanding, even if the application was not built using WebSphere Studio Enterprise Developer,

Changes for VisualAge Generator Programmers

Since WebSphere Studio Enterprise Developer is based on the WebSphere Studio Application Developer (WSAD), there will only be a Java version of the product. **There is no Smalltalk version of WebSphere Studio Enterprise Developer planned.** Today you can migrate everything from the Smalltalk-based version of VisualAge Generator to the Java-based version, excluding the GUIs. A VisualAge Generator business partner is currently exploring options to provide the needed GUI migration.

Finally, the WebSphere Studio Workbench yields a development environment oriented towards Windows and Linux programmers. There are no plans for an OS/2 ® version of the Workbench, so VisualAge Generator users with OS/2 for their IDE will have to migrate to another development platform.

Migrating to WebSphere Studio Enterprise Developer

WebSphere Studio Enterprise Developer will consist of an Integrated Development Environment (IDE) built upon on the WebSphere Studio Workbench, and a WebSphere Studio Enterprise Developer run-time. The run-time will also be able to run Cross System Product and VisualAge Generator applications, for all platforms supported by WebSphere Studio Enterprise Developer.

Migration tooling will be provided to transform the VisualAge Generator 4GL into WebSphere Studio Enterprise Developer's EGL. There will also be tools to help migrate code from the Envy-based repositories of VisualAge Generator to the user-selected repositories supported by the WebSphere Studio Workbench. The WebSphere Studio Enterprise Developer run-time will be capable of executing programs generated by Cross System Product 4.1, VisualAge Generator, and WebSphere Studio Enterprise Developer.

More Information

You can find the latest VisualAge Generator news in one of three ways:

- Access product newsletters on the VisualAge Generator web site: http://www.ibm.com/software/ad/visgen. Newsletters are first published on the web, and then distributed in a printed document. You can sign up at the web site to receive printed copies of the newsletter. On the web site, for example, you can find the recently published article explaining how VisualAge Generator web transaction functionality and an IMS database were used to track a fleet of vehicles at the Commonwealth of Kentucky (http://www-4.ibm.com/software/ad/visgen/library/).

- Follow the VisualAge Generator product newsgroup at news://news.software.ibm.com/ under ibm.software.vagen. You can use the newsgroup to communicate with the VisualAge Generator lab, other customers, and VisualAge Generator business partners.

- Attend a VisualAge Generator conference in 2002. At the conferences, you will hear about the latest releases, as well as have the opportunity to compare notes with other VisualAge Generator customers and business partners.

  - VAG Asia Pacific Users Group (now part of Interaction Australasia)
  - Nordic GUIDE
  - SHARE – several conferences held around the world
  - VAGen Users Group
  - VAG Germany Users Group
  - VAGen International Symposium
  - WebSphere Summit – several conferences held around the world

You can keep current on products based on the WebSphere Studio Workbench by watching the Workbench web site –http://www-4.ibm.com/software/ad/workbench/. Currently WebSphere Studio Enterprise Developer does not have a web site since specifics of this product have not been announced.

Conclusion

As we hit the end of this paper, there are a number of points worth re-emphasizing:

1. Cross System Product and VisualAge Generator have evolved with changes in the industry, incorporating new architectures, data stores, communication protocols, and web technologies. Our focus has never been on incorporating wizards and widgets to use the latest technology just for the sake of the technology; it has been on leveraging the skills of current development staffs and enabling them to rapidly develop applications that will run today's largest companies.
2. The most recent releases of VisualAge Generator (V4.0 and V4.5) can facilitate the creation of complete web solutions running under application servers such as the WebSphere Application Server.
3. VisualAge Generator usage has seen significant growth in 2001 and 2002, in spite of an economy that is sluggish.

4. Application Development customers have repeatedly made requests for additional functionality, integration with other application development products (IBM and ISV-supplied), the ability to use alternate repositories, and access to the latest Internet technology for building web applications.
5. IBM responded to those requests with the new WebSphere Studio products, based on the WebSphere Studio Workbench. The Workbench provides an integrated foundation for the development of applications by IBM, ISVs, and business partners. All tools that take advantage of the Workbench will have a common look and feel, widget tool kits, source editing and graphical frameworks, and access to user-selected repositories.
6. WebSphere Studio Enterprise Developer is the Workbench-based product for VisualAge Generator, VisualAge COBOL, and VisualAge PL/1 customers. This is a integrated enterprise development tool for 4GL (now EGL), Java, COBOL, PL/1, and web programmers.
7. The initial release of WebSphere Studio Enterprise Developer is focused on building web applications. Subsequent releases will include more Cross System Product and VisualAge Generator functionality and platforms. Migration tooling will be provided from Cross System Product and VisualAge Generator to WebSphere Studio Enterprise Developer as the corresponding functionality is added to WebSphere Studio Enterprise Developer.
8. IBM will extend the end-of-service dates for VisualAge Generator and its co-requisite and pre-requisite products with a new V4.5.4 release of the product.
9. If you are a Cross System Product customer or are on an unsupported version of VisualAge Generator, don't wait for WebSphere Studio Enterprise Developer; move to VisualAge Generator V4.5 today and immediately take advantage of its rich capabilities to build TUI, client/server, and web applications.

With Cross System Product and VisualAge Generator, IBM has provided over 20 years of tooling to help the world's largest enterprises rapidly develop and support mission-critical applications. With the WebSphere Studio Enterprise Developer, IBM continues this rich tradition in providing an integrated enterprise-class tool that takes your core business applications into the exciting world of dynamic e-business.

If you have any questions regarding this paper, VisualAge Generator, or WebSphere Studio Enterprise Developer, feel free to e-mail Steve Choquette, the VisualAge Generator Product Manager, at choquet@us.ibm.com.

Acknowledgements

The author would like to thank the following people whose earlier papers I shamelessly borrowed from to produce this document: Joe Pesot, Stephen Hancock, Cliff Meyers, Tim Wilson, Larry England, and Rusty Edmister.

In addition, I received numerous suggestions from Stefano Sergi and Daphne Green, which were all valuable.

Notices

The following terms are trademarks or service marks of the IBM Corporation in the United States or other countries or both: AIX, AS/400, CICS, CICS OS/2, COBOL, DB2, IBM, IMS, MQSeries, MVS, VM, VSE, OS/2, OS/390, OS/400, RISC System/6000, WebSphere, VisualAge, and VisualGen.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Microsoft, Windows, Windows NT, Windows XP, and Windows 2000 are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and services names may be trademarks or service marks of others.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information is for planning purposes only. The information herein is subject to change before the products described become available.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice