Application Infrastructure & Software Platforms
by Dana Gardner
June 2004

Graphical RAD Tools for Java Align Application Demand with Developer Supply

Executive Summary

Enterprises rightfully have been seduced by the virtues of thin-client computing over the past 6 years. By now even laggards exploit Web browsers, IP networks and server-based intranet applications and portals to offer their employees swift access to data.

Exposing applications and data to the Web user interface is a simple way to begin breaking down the boundaries of monolithic applications and expensive client/server application dependencies. Two-tier, thin-client applications have made essential information readily available and cheap.

In response to the massive need for Web-tier views into applications and data, developers and scripters emerged with a slew of various tools and methodologies. The productivity benefits enabled by browser-based IT overrode concerns about the best ways to manage end-to-end distributed computing.

However, the thrill of the intranet is now passé. Enterprises must better address costs and complexity through a more thoughtful approach to the full spectrum of applications and platforms. More complex logic-based application development, such as Enterprise Java, should no longer be divorced from presentation-level activities such as scripting and data connecting. Enterprises now must align their total application development demands with the best-of-breed architectures and frameworks.

Enterprises want to cut total costs by better coordinating their development capabilities with their strategic architectural goals; improve the productivity of their developer corps; and sensibly reduce the complexity of their application portfolios and supporting platforms. Enterprise leaders tell us they want to standardize on the platforms that provide the best value, performance and flexibility; reduce the number of tools they use; and securely and reliably move to more productive IT innovations such as Web services and services-oriented architectures (SOAs).

However, there needs to be a shift in how developers and tools are matched to allocate and train developers in ways that make the most sense in the long term. Trends in grid/utility computing, standardized development frameworks, and applications and services modeling are accelerating the need for this shift.

Given the forces at work in the marketplace, it is no surprise that the use of many new rapid application development (RAD) tools for Java is growing. These tools include:

- BEA WebLogic Workshop
- Borland JBuilder
- Compuware OptimalJ
- Emacs Editor
- IBM Rational and WebSphere Studio
- Microsoft Visual Studio
- Oracle JDeveloper
- SAP NetWeaver
- Sun Java Studio Creator

The arrival of these visual tools marks an important new level of maturation in the evolution of development. However, the most successful tools are those that not only address technical requirements, but also align the business-driven demands of tomorrow's most productive application projects with a shifting supply of developers.

With graphical RAD tools for Java, there comes a rare opportunity to align available skills, appropriate tool characteristics, and the needed outcomes for application function and implementation. By blending the best aspects of several kinds of tools—from graphical drag-and-drop functionality to the strength of the Java deployment paradigm—the focus of tool performance has shifted from purely technically defined goals to organizing developers in a more productive way. Graphical Java tools for Web-interface applications now help bring together the right skills with the right kinds of simplified applications on the right platforms.

With an added emphasis on developer productivity, Java tools will improve how enterprises transition between frameworks, skill sets and the most sought after data access interfaces. The Yankee Group predicts Java will further its dramatic expansion of the past 5 years. For many enterprises, expanding the role of Java tools generates swifter ROI on their Java-compliant middleware and platforms, and brings more developers under a common framework for higher productivity and team-oriented management.

However, simplified RAD tools and associated methods vary widely. The many goals assigned to these new tools and suites are so complex that choosing the right path will be difficult for enterprise architects. This report examines the forces behind the creation of new RAD tools for Java, how the developer community will be affected, and how Sun Microsystems Inc.'s Java Studio Creator tool meets enterprises' developer resources management needs.

Table of Contents

I. Introduction

There seem to be as many tools today as there are types of applications. Yet over time, the tools that remain in popular use do not necessarily keep pace with application requirements. The use of outdated tools can also sap developers of their productivity potential by keeping them from using newer tools on newer frameworks, or from joining teams that are conducting more complex and fruitful projects.

Now is the time for enterprises to reevaluate their tools strategy as part of a larger shift to leverage unified frameworks, organize applications under portals and exploit SOAs.

There are several reasons why this is an auspicious time for such a reevaluation:

- During the past few years, a gap has emerged between the business requirements for IT departments and how those shops have been organized. Many development groups remain organized in the way that supported host or rich client-server-oriented projects.
- Large monolithic applications have given way to highly complex, large distributed applications. Java 2 Enterprise Edition (J2EE) and its associated IDEs have grown dominant in addressing such projects.
- Two-tiered server-based applications and portals with Web user interfaces have taken the place of client/server applications. Older fourth-generation languages and Microsoft Visual Basic as well as scripting languages have been repositioned to the Web paradigm with mixed results.
- Many enterprises must juggle at least two major development paradigms—Web and distributed—each with separate cultures, separate tools and often different deployment platforms.

It is not ideal to support two sets of deployment frameworks and platforms because the prevalent tools for certain types of applications demand differing runtimes. It makes more sense to bring the right tools that align the available skills with the predominant runtimes. It makes even more sense if the Web applications form the top tier of a larger interoperability framework such as Java.

Since a Java-based runtime environment is in place at most enterprises, wouldn't it make sense to leverage that same environment for Web applications and portals? Shouldn't the development teams be aligned? Why make million-dollar deployment infrastructure decisions based on a developer skill fit when the skill fit can be tailored to the application need on the most flexible platform?

The answers seem obvious. However, a key ingredient has been missing: the ability to take developers—whether COBOL programmers from legacy systems, third-generation-language developers such as Fortran, PowerBuilder developers from rich client/server architectures, or Visual Basic developers and scripting developers from Web applications—and redirect them appropriately and productively.

Graphical RAD tools for Java are emerging to align developers from outmoded design and deployment paradigms to an increasingly Web-tier, portlet- and services-oriented world of end-to-end distributed computing. Ideally, the same teams—within the same frameworks—will produce the many elements that come together to form most applications and XML-based services, which by definition can consume the outputs of previous applications regardless of native platform.

The Yankee Group estimates that globally through 2007, between 4 and 7 million developers will change the tools they use for most of their work. This significant shift is based on a number of factors including the continued strong growth of Java tools, consolidation of applications and platform types, and the growing need for higher level modeling-based and business-process-based development projects.

Better aligning application demand with developer supply will yield more developer jobs and teams that are more productive. Their employers—who must now make some important decisions about how many tools and frameworks make sense, both functionally and fiscally—also will benefit.

Although many of these developers undoubtedly will climb the skills ladder by becoming proficient in coding-intensive Enterprise Java and Microsoft .NET architectures, many more will make lateral moves from one environment to another, or from a scripting or host language, for example, to a visual environment that is easier to learn and has the most long-term use.

Enterprises should recognize this rising churn in the developer ranks now. They should plan by making choices about the platforms of highest value, their most productive application modeling methodologies, and the means to gain and hold the proper developer skills sets to support these shifts.

II. Time for a Real Change

Developers currently use many tools to create simple, two-tiered server-based applications and portals that usually connect to structured data, XML and Web services. Scripting tools such as Perl and JavaScript remain popular. For Windows server deployments, Visual Basic and its kin within Visual Studio are very popular.

However, to prepare for developer resources management, a leapfrog effect is needed to bring more developers into the role of end-to-end-solutions-based—rather than platform-based—development. These developers can then join the teams of J2EE developers to more quickly produce cross-platform solutions that span more device interfaces and many more data and object sources. These teams will bridge backward and forward compatibility of programming in general.

How should enterprises forge these new teams? Where will the developers come from? Several hundred thousand COBOL programmers, for example, need a path to working within the predominant Java frameworks without mastering Enterprise Java APIs. Visual Basic developers—seeking to play in the larger solutions-oriented marketplace rather than be limited by the Windows-only marketplace—will want to use their skills to deploy more widely. Many object-oriented programmers will want to modernize for SOAs.

Additionally, Yankee Group analysis shows that the largest segment of developers in the United States and Canada are C/C++ developers, with a budding cadre of Microsoft C# developers in tow. Many Visual Basic developers are already considering a shift to C# and to the Microsoft .NET framework. If they are preparing to make such a dramatic shift, they may consider a shift to graphical RAD tools for Java first as a segue. From there they can direct their new skills to a wider potential addressable market. There must be a progression from several legacy tool skills sets to a graphic- and modeling-oriented tool set that adheres to a Java framework and supports SOAs.

However, enterprises that are expensively supporting many groups of developers will want to better organize them to better modernize their skills and align their training with common deployment scenarios and goals. These goals include swift project completion on currently installed platforms that have demonstrated reliability, security and scalability.

There remains a shortage of expert Enterprise Java developers worldwide. Aligning the teams through developer resources management eases challenges with data- and services-tier interfaces and prepares many more developers to tackle higher-level Java and SOA architectural programming and business logic design tasks.

To be successful, graphical RAD tools for Java should promote all of these goals and provide a path to the future by doing more to consolidate the disparate enterprise development corps.

What Creator Brings to the Table

Sun Microsystems has proposed a phased approach to its new tools strategy. The first tool, Java Studio Creator, will debut in the summer of 2004. Based on NetBeans, it exploits JavaServer Faces (JSFs), a specification recently approved by the Java Community Process. JSFs ease the building of server-side UIs without the need to delve into J2EE and Struts.

By including Java framework-based standards, Java Studio Creator enables drag-and-drop assembly and visual benefits that automate tasks and hide Java intricacies, opening Java-deployed development to a far wider audience. The approach uses a JSF drag-and-drop methodology for point-and-click access to connect to such functional tiers as relational databases and Web services.

By automating some connectivity elements to the Web servers, developers can quickly build simple two-tier applications or portals that deploy to a Java virtual machine or Sun Java System Application Server 7 without the need to know low-level details of J2EE APIs, model view controllers or Struts. The approach relegates JavaServer Pages (JSPs), servlets and EJB development to a separate but related undertaking within a Java IDE.

In essence, with Creator, the middleware connectivity elements below the interface tier either are automated or are relegated to the tasks best suited to a J2EE-class developer and most sophisticated tools. The segmented approach, adjusting to the realities of the available skill sets, fills out the development suite for Java and enables the manageable stratification of Java design and deployment project elements.

For many Java developers, Struts has emerged as a favored methodology for Web-tier applications. However, Sun is intent on delivering JSFs as a means to expand productivity and resources management for a wider class of developers. Sun is seeking a leapfrog strategy for tools by attracting developers from the ranks of legacy tools and Windows-only visual tools who can deploy Java runtimes. In addition, Sun is working with component vendors to have supplementary JSF components available for use with Java Studio Creator and other tools that support the standard.

Sun is pricing the tool aggressively, with subscriptions through the Sun Developer Network as low as $99 per year. The goal is to build a path to a number of forthcoming Java simplification tools. The next release, code-named Thresher, will arrive by the end of 2004 and will include one-button deployment of portlets to portal servers, a wider range of application servers, and code management features. The third release, code-named Mako and due in the second half of 2005, will support multiple and rich clients, Swing interfaces, Web services constructs, session beans and increased modeling capabilities.

Sun is also working with component vendors to embrace Unified Modeling Language (UML), an Object Management Group-sponsored modeling specification, through third-party partnerships. The UML 6.1-enhanced tools will come out with Java Studio Enterprise later in 2004. Sun is working with component vendors to have supplementary components available for use with Java Studio Creator.

Making Java More Productive

Many enterprises have aggressively moved to intranet portals for internal aggregation of applications to enhance the availability of data across an organization's business groups. Many of these enterprises also are moving away from tight coupling of specific portal tools with proprietary portal servers to a more infrastructure- and standards-based approach relying on portlets.

The Yankee Group estimates graphical RAD tools for Java such as Java Studio Creator can accommodate up to 50 percent of new portal- and intranet-based Web interface application development in such organizations. These are companies that want to quickly assemble Web sites or intranet Web applications without the high costs of managing disparate frameworks. They also seek to reap the benefits of developer resources management in the process.

These thousands of enterprises—often with dozens of Web applications each—should evaluate how they construct and deploy Web applications internally and externally. When enterprises consider the benefits of developer resources management and skills alignment, as well as the ability to further exploit an existing or new Java-based deployment architecture, the potential improvements in productivity offered by RAD tools for Java are impressive.

For vendors of general Java IDEs, which tend to favor a J2EE-class developer, the suite approach may be overkill for the benefits of simple Web interface views of data. Smaller companies or IT shops with limited budgets may not want or cannot afford the fuller support of a BEA, IBM or Borland development approach for the Java-based two-tier applications.

The tight integration between Microsoft tools and the Windows Server System, while often highly productive in closed environments, can prove costly for producing heterogeneous portals due to the number of products, upgrades and licenses involved. Security concerns and the need to support various cadres of developers can add to total costs.

With the Java Studio Creator tools set, Sun is carving out a needed niche within enterprise development circles—one that has much broader implications than its immediate functional capabilities. RAD tools that can streamline a developer core and make it more highly focused and therefore productive—and can align application demand with developer supply across a flexible framework—offer the more important and cost-effective benefits.

Sun's future tool releases will accelerate these benefits and extend them to the development of rich client-facing applications, as well as support far greater numbers of devices such as mobile handsets and microdevices. The goal is to reduce the complexity and costs of end-to-end development functions by allowing far more developers to exploit the benefits of Java.

III. Conclusions and Recommendations

The test for measuring developer tool performance has shifted. In addition to considering purely technically defined goals, enterprises also consider the role of skills transference and the developer corps in the organization.

The Yankee Group estimates that through 2007, between 4 and 7 million developers will change the tools they use for most of their work. Enterprise architects and decision makers will need to decide what developer competencies will pay the biggest return and insulate them from the risk of a lingering discrepancy between application demand and developer supply. With graphical RAD tools for Java, there comes a rare opportunity to align available skills, appropriate tool characteristics, and the needed outcomes for application function and implementation. However, the many goals assigned to these new tools and suites are so complex that choosing the right path will be difficult for enterprise architects.

With the Java Studio Creator tools set, Sun Microsystems is carving out a needed niche within enterprise development circles—one that has much broader implications than its immediate functional capabilities. Sun is seeking a leapfrog strategy for tools by attracting developers from the ranks of legacy tools and Windows-only visual tools who can deploy Java runtimes. Graphical RAD tools for Java such as Java Studio Creator can accommodate up to 50 percent of new portal and intranet-based Web interface application development.

The segmented approach of Creator, which either automates or relegates tasks best suited to a J2EE-class developer and most sophisticated tools, adjusts to the realities of the available skill sets, fills out the development suite for Java and enables the manageable stratification of Java design and deployment project elements. Although Sun Java Studio Creator targets the corporate developer, it also will be of interest to ISVs and hosted services providers.

With an added emphasis on developer productivity, Java tools will improve how enterprises transition between frameworks, skill sets and the most sought after data access interfaces. The Yankee Group predicts Java will further its dramatic expansion of the past 5 years. For many enterprises, expanding the role of Java tools generates swifter ROI on their Java-compliant middleware and platforms, and brings more developers under a common framework for higher productivity and team-oriented management.

Recommendations to Enterprises

- Recognize the imminent churn in the tools developers will use and make choices about the frameworks and platforms of highest value in the long term.
- Evaluate how you construct and deploy Web applications internally.
- Encourage architects and developers to evaluate their skill sets and how they expect to meet their future development needs within these frameworks.
- Examine how cultural shifts in methods of development and deployment can increase the value of IT while decreasing total costs over an application's life cycle.
- Emphasize ease of development when choosing tools and deployment frameworks to expand choices on developer hiring and outsourcing.
- Explore and rate various tools based on speed, cost and simplicity.
- Learn more about graphical RAD tools for Java.

IV. Further Reading

Yankee Group Application Infrastructure & Software Platforms Reports

New Hope Emerges for Ending the Costly Disparity Between Java and Windows, June 2004

Next 24 Months Will Bring Testing of Web-Services Toolkits, Tools, and Networks, June 2003

Yankee Group Application Infrastructure & Software Platforms Research Note

Sun Takes Java to the Next Level, March 2004

The Yankee Group

World Headquarters

31 St. James Avenue
BOSTON, MASSACHUSETTS 02116-4114
T 617.956.5000
F 617.956.5005
info@yankeegroup.com

Regional Headquarters

North America

31 St. James Avenue
BOSTON, MASSACHUSETTS 02116-4114
T 617.956.5000
F 617.956.5005
info@yankeegroup.com

951 Mariner's Island Boulevard, Suite 260
SAN MATEO, CALIFORNIA 94404-5023
T 650.522.3600
F 650.522.3666
info@yankeegroup.com

EMEA

55 Russell Square
LONDON WC1B 4HP
UNITED KINGDOM
T 44.20.7307.1050
F 44.20.7323.3747
euroinfo@yankeegroup.com

For More Information

T 617.956.5000
F 617.956.5005
E-mail: info@yankeegroup.com
Web site: www.yankeegroup.com

Advisory Services

Yankee Group advisory service annual memberships offer clients access to research and one-to-one expert guidance.

Advisory services represent our best value for clients. The services help our members understand industry, regulatory, competitive and market-demand influences, as well as opportunities and risks to their current strategies.

Membership includes an invaluable in-person strategy session with Yankee Group analysts, direct access to a team of analysts, research reports, forecasts, research notes and regular audioconferences on relevant topics.

We offer advisory services on almost 30 selected topics in Telecommunications; Wireless/Mobile Communications; Consumers, Media & Entertainment; and Information Technology Hardware, Software & Services.

Decision Instruments

The Yankee Group offers a full portfolio of technology and market forecasts, trackers, surveys, and total cost of ownership (TCO), return on investment (ROI), selection and migration tools. Decision instruments provide our clients the data required to compare, evaluate or justify strategic and tactical decisions—a hands-on perspective of yesterday, today and tomorrow—shaped and delivered through original research, in-depth market knowledge and the unparalleled insight of a Yankee Group analyst.

Trackers
Trackers enable accurate, up-to-date tactical comparison and strategic analysis of industry-specific metrics. This detailed and highly segmented tool provides discrete proprietary and performance data, as well as blended metrics interpreted and normalized by Yankee Group analysts.

Surveys
Surveys take the pulse of current attitudes, preferences and practices across the marketplace, including supply, delivery and demand. These powerful tools enable clients to understand their target customers, technology demand and shifting market dynamics.

Forecasts
Forecasts provide a basis for sound business planning. These market indicators are a distillation of continuing Yankee Group research, interpreted by our analysts and delivered from the pragmatic stance our clients have trusted for decades.

Signature Events

The Yankee Group's signature events provide a real-time opportunity to connect with the technologies, companies and visionaries that are transforming Telecommunications; Wireless/Mobile Communications; Consumers, Media & Entertainment; and Information Technology Hardware, Software & Services.

Our exclusive interactive forums are the ideal setting for Yankee Group analysts and other industry leaders to discuss and define the future of conversable technologies, business models and strategies.

Consulting Services

The Yankee Group's integrated model blends quantitative research, qualitative analysis and consulting. This approach maximizes the value of our solution and the return on our clients' consulting investment.

Each consulting project defines and follows research objectives, methodology, desired deliverables and project schedule. Many Yankee Group clients combine advisory service memberships with a custom-consulting project, enabling them to augment our ongoing research with proprietary studies.

Thousands of clients across the globe have engaged the Yankee Group for consulting services in order to hone their corporate strategies and maximize overall return.

www.yankeegroup.com