IDC
Analyze the Future

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

WHITE PAPER

Four J's: Evolving Applications for SOA

Sponsored by: Four J's Development Tools

Sandra Rogers

August 2006

ABSTRACTING COMPLEXITY

Today's organizations need to make the most of their IT resources. But if applications are too closely designed to specific operating systems, deployment environments, databases, user interfaces, and client devices, the value and flexibility of those solutions can be limited. Enterprises have struggled for years to connect systems, especially when they are of different makes, and even generations, of technology. Many organizations have attempted to rein in these efforts through various means, often resorting to point-to-point integration approaches or consolidation efforts. Whether to leverage and extend existing systems, or perhaps port applications for use on varied environments, the need to create an architecture and foster development in a uniform way that can address many alternative system configurations is certainly a more desirable tack.

Market interest, industry standards, and technology innovations are advancing to support architectures that promote greater levels of abstraction, allowing for components of systems to be created and exist independently. If implemented appropriately, such design approaches and assets can contribute to greater efficiencies and create systems that more readily adapt to change.

Service Oriented Architecture and Web Services

To address key IT goals of standardization, interoperability, and reusability to help address business needs including time-to-market pressures, competitive differentiation, compliance, and cost control, enterprises of all size classes and industries are embracing the concepts of system and application virtualization and dynamic connectivity. These desires converge with the overwhelming market trend to leverage centralized, shared IT resources supplied by both internal and external parties. To enable this, it is becoming more pertinent to utilize system environments and industry-standard protocols and allow for greater connectivity. As testimony to this trend, service oriented architecture (SOA) and Web services continue to amass tremendous market interest and expanding levels of adoption.

Briefly defined, SOA promotes the utilization of autonomous application and system "services" abstracted from one another, independent of implementation. Ideally, an SOA should be modular, with separate layers of functional code, data, workflow, and presentation interfaces. Each service should be self-describing, with a published interface, and accessible to other elements of the system, most commonly over a network. In an SOA, services are designed to be dynamically invoked.

Ideally, the preference is to natively design systems to leverage layers of the application infrastructure as services themselves and create effectively defined and bounded services. However, the ability to capitalize and extend existing systems by enabling them in a services capacity is a key driver for many organizations adopting SOA. Thus, what had been previously relegated to proprietary extensions and customization (and the need to employ dedicated system resources and expertise to support them) can then be open to a whole new world of possibilities.

The concept of leveraging modular, distributed code and application programming interfaces (APIs) is not new. Past attempts to use these methods were primarily proprietary in nature, requiring investment and maintenance of transformational code in the form of adapters, customized scripting, and investment in compatible infrastructure. With growing industry support for XML and Web services standards, the ability to address applications that have been enabled or natively designed utilizing these protocols is easing the complexity. This is accomplished by normalizing some of the basic operational syntax required to interact with these interfaces.

Standards-based SOAs can be extremely useful for organizations that are challenged by highly heterogeneous and volatile IT environments. Standards continue to evolve, layering upon basic formats the aspects of reliable messaging, identity, security and management policy, optimization, and other processing dynamics related to communications between systems. Examples of key technology-oriented standards include XML, SOAP, WSDL, WS-Security, WS-Reliable Messaging, WS-Policy, BPEL, and more. Some of the more well-known business and industry standards include RosettaNet, UCCnet, CIDX, HL7, and SWIFT, among the countless niche industry variations based on XML.

A Separation of Concerns

The primary draw of distributed systems is that one component or tier can theoretically exist and be modified without necessarily physically impacting all other elements of the system. With such design, standardized, shared systems and software can be more realistically leveraged and replaced as needed. Each module of the system depends, however, on proper access to and understanding of the interface of the other parts of the system.

Typically, a three-tier client/server model involves the separation of tiers of code dedicated to the user interface, the business logic, and the information/data. N-tier then extends this concept, adding Web serving and other intermediary processing capabilities as needed. SOA is complementary to these traditional distributed approaches in application design, adding dynamic access, invocation, and management of information and functional assets as "services." Each service is designed as an independent entity and should perform a cohesive and well-defined set of functions without knowledge of the consuming service or any underlying infrastructure or service dependencies.

The ability to route messages and trigger services in a dynamic fashion utilizing abstracted operational and business rules is a key technological foundation supporting flexible use of services and is the area where many platform and process vendors are focusing efforts to support coordination and intermediation of event-based messages and quality-of-service (QoS) activities. By creating levels of abstraction in the core application design model, enterprises can more easily and cleanly define, interject, enforce, and monitor processes and policies. IT capabilities to support SOA include integration, process orchestration, state management, policy, and deployment. Surrounding these capabilities are IT services that determine how solutions are built (development), accessed by people and other environments (access), secured and managed (security and management services), and utilized (application and data services). What specific application capabilities are targeted to be provisioned as services, how neatly they are created, and how effectively they are constrained and defined — and more — all contribute to the overall effectiveness of an SOA environment.

Many organizations are creating services from existing applications and combining these with newly formed processes and solutions. Increasingly, discrete content and functionality provided by third parties are also being enveloped into the mix of services. Enabling specific modules of solutions as services and building up repositories of available enterprise-sanctified services are keys to moving forward in the SOA world.

Architectures that promote abstracted layers can allow significant flexibility; however, issues of performance, reliability, and complexity must be considered. What most organizations are seeking is the right balance between these concerns. These efforts are also weighed against the need to address time-sensitive business requirements versus longer-term IT strategies and policies. An SOA by design should and will evolve; thus, the ability to incrementally add to and finesse one's overall solution needs to be addressed.

FOUR J'S: DEVELOPING FOR CHOICE

Four J's is a software company that for over a decade has steadfastly focused on creating an application development environment that is open and flexible. The company, founded in 1995, has evolved from providing a dedicated 4GL Informix-based solution to support a variety of platforms, information sources, and interface styles in its rapid application development (RAD) environment. Four J's architecture embraces the concepts of abstraction for greater levels of flexibility and interoperability.

Four J's has a global presence with European headquarters in Shannon, Ireland, and North American headquarters in Irving, Texas. The company has 14 offices across the United States, Europe, Asia/Pacific, and Latin America. The firm has built up a sizable customer base of end-user organizations, ranging from Fortune 500 enterprises to medium-sized businesses, public agencies, and technology-based firms. The company reports that it has more than 2 million user licenses and a community of 5,000 developers worldwide, including its large ecosystem of value-added resellers (VARs). Supporting this demanding audience, Four J's has concentrated on easing the burdens of navigating and optimizing the tiers of data and interface access, enabling its clients to focus their own development efforts on the application logic that adds value and differentiation to their businesses.

Many VARs typically focus on addressing very specialized requirements for certain industries or particular use cases. They often are also key providers of technology to small and medium-sized businesses (SMBs) that require foundational application infrastructure and reporting capabilities to be integrated into their solutions. Similarly, enterprises that use 4GLs are looking for an environment that can readily evolve with changing business requirements.

What one must consider, however, is that over time, the Four J's environment has advanced to support systems that bridge varied generations of technology, allowing mission-critical solutions to evolve as needed. Although Informix is heavily associated with Four J's, the company has embraced a cross-platform philosophy that has made it an invaluable resource to much of its diverse and loyal customer base. To point, this private firm reports sustained steady revenue growth and profitability levels and that it continues to expand its reach into new market segments and geographies. The company's VAR partners that were interviewed for this paper show healthy growth and pipelines within their respective businesses.

Four J's Genero

Four J's most recent product offering, Genero, further optimizes the use of a tiered development architecture and design methodology. This is the latest version of Four J's development platform that fosters the creation of screen design and database interaction separate from core application business logic.

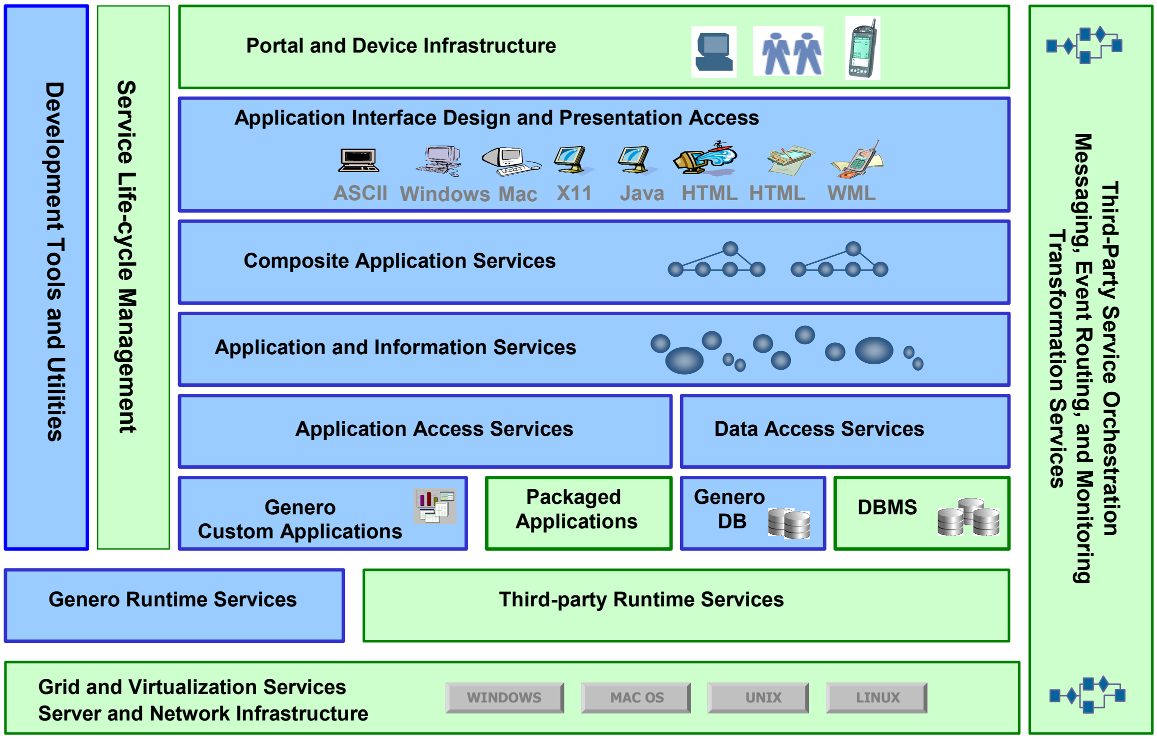
Four J's uses XML to create a uniform presentation layer, thus freeing the developer from needing to understand and code to graphical user interfaces (GUIs) for specific and multiple clients. The Genero offering has also been updated to automate the creation of Web services for more rapid integration and construction of business solutions. Four J's is dedicated to advancing its Web services and industry protocol adherence as market demand warrants.

Four J's also supports a variety of database environments. Applications developed on (or ported to) Genero can call upon a variety of native database drivers and run on a mix of operating systems, including UNIX, Linux, and Windows. In 2006, the company released its own database offering, Genero db, incorporating ANTs data server technology "under the covers." It is a high-performance relational database with replication and failover capabilities for automatic recovery and online backup.

Figure 1 highlights (in blue) at a high level where the Genero solution offerings can contribute to an overall application and SOA landscape. The company continues to evolve its capabilities and adherence to standards that will be most critical in a shared and heterogeneous solution environment.

FIGURE 1

Four J's Genero: Contributing to an Evolving SOA Architecture



Development Tools and Utilities
Service Life-cycle Management
Portal and Device Infrastructure
Application Interface Design and Presentation Access
ASCII
Windows
Mac
X11
Java
HTML
HTML
WML
Composite Application Services
Application and Information Services
Application Access Services
Data Access Services
Genero
Custom Applications
Packaged
Applications
Genero
DB
DBMS
Genero Runtime Services
Third-party Runtime Services
Grid and Virtualization Services
Server and Network Infrastructure
WINDOWS
MAC OS
UNIX
LINUX
Third-Party Service Orchestration
Messaging, Event Routing, and Monitoring
Transformation Services

Note: Portions where Four J's Genero contributes are highlighted in blue.

Source: Four J's Development Tools Ltd. and IDC, 2006

The primary elements of the Four J's Genero offering include:

- **Genero Business Development Language (BDL).** Four J's business process grammar is optimized at a metalevel to facilitate business logic programming, database queries, and user interface design.
- **Genero Studio.** This integrated development environment (IDE) includes an application generator that autocreates source code for rapid development.
  - ❑ **Genero Studio Graphical Debugger.** The debugger enables real-time source-code scrolling during program execution in a separate text, Microsoft Windows, or HTML client window. The tool supports onsite and remote debugging, XML and XSL preprocessors, and CGI testing.
- **Genero Dynamic Virtual Machine (DVM).** The application server runtime environment establishes a common protocol link with each client type based on XML, managing and optimizing communications between client and server. Program source code is compiled to an intermediate level, which is processor, operating system, database, and user interface independent, thus allowing it to execute across a wide variety of platforms without recompilation. Business logic and database transactions reside on the application server, which is separate from the Web server that forwards data streams and presentation information to the browser. This helps further modularize the activities of the application and allows for added security and process-flow options.

- **Web services standards and autogeneration of schema for publishing and consuming Web services.** As of the Genero v2.0 release, Four J's supports SOAP version 1.1 and WSDL version 1.1, including document literal style Web services and Microsoft .NET service types.

- **Genero Desktop and Web Clients — Abstract User Interface (AUI).** Client user interfaces are built with XML that is abstracted from the business logic. Presentation design facilities include graphic style sheets and autoconfiguration for varied desktop and Web clients requiring no recoding or recompiling of source code. This element includes support for Windows, Java, ActiveX, HTML, ASCII, X11, and WML (for WAP).

- **Genero Open Database Interface (ODI).** The Genero ODI provides native drivers for most databases, including Genero db, Informix, IBM DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, and Sybase. Four J's indicates that its architecture allows one to connect to multiple database engines simultaneously.

CASE STUDIES

IDC interviewed two Four J's customers to glean further insight into the company's product offerings and user experiences. In-depth phone interviews were conducted with critical personnel associated with the selection and use of the technology.

Tyler Technologies' MUNIS

Tyler Technologies is a United States–based holding company focused on providing business information management solutions to local and midsize government entities, schools, utilities, and nonprofit agencies. Tyler's MUNIS division, currently serving over 1,000 customers, sells an enterprise resource planning (ERP) solution including finance, HR/payroll, productivity, and ebusiness application modules specifically designed for this industry group. The company offers its software in traditional licensing form and is growing a base of customers for its application service provider (ASP) hosted solution.

Many MUNIS clients are in a challenging position when it comes to procuring new systems and functionality because they are often resource constrained and have critical dependencies on legacy systems. They are also typically risk averse, not wanting to disrupt their own respective constituent bases. MUNIS has the opportunity to supply these organizations with a prudent and compelling migration path to adopt and leverage newer technologies.

The competition from top-tier packaged application providers places significant pressure on MUNIS to consistently add deeper functionality in a fully integrated offering. The company uses Genero as its core development environment and foundation for its ERP package. Enabled by Four J's solution, the same code can now run in a Windows or browser client due to its abstracted XML-based presentation layer, allowing the company to offer a full Web browser–based solution with a zero-client install. This is important for MUNIS customers to extend more self-service functionality to a distributed base of users, thus cutting down on administrative work.

Add-on products sit on top of the MUNIS Four J's platform, built with Genero or other development tools and technologies. One example is that MUNIS is using Microsoft .NET, Visual Basic, and C++ to enhance specific self-service functions aimed at helping its clients extend core ERP solutions to their constituent citizens.

Greg Savard, vice president of R&D at MUNIS, and his team are responsible for making sure the company has a competitive product in the marketplace and that it is regularly adding the most important features and functionality to meet demand. Part of this group's role is to also regularly evaluate what types of technology are available in the market for MUNIS to leverage, perform assessments, and set strategic architectural direction that subsequently rolls out to the greater development population in the division.

The firm started using Four J's in the mid-1990s as a stopgap measure to move from its old 4GL character-based system to a GUI-based paradigm. However, what was first viewed as a short-term approach while the company investigated rewriting its systems soon became MUNIS going beyond opening up access to the Informix back end and moving to support three different databases with this environment, including Informix, Oracle, and Microsoft SQL Server with a single source stream. This tack became apparent after seeing the flexibility that the Four J's product offered. According to Savard, adopting Four J's and Genero has allowed MUNIS to focus on building out or plugging in new additive functionality. By providing a mechanism to coordinate the data access layer and retain and protect the business logic layer, MUNIS developers can be separately focused on advancing the user experience.

The development team first started down the path of creating its own Web services using a combination of technologies such as COM, ASP, and ODBC and quickly found it was duplicating code. Once Four J's added Web services capabilities, MUNIS quickly jumped on board to take advantage of the efficiencies it afforded. Now, service-enabling its ERP solution and supporting Web services standards with its latest solution allows MUNIS customers more flexibility to build out some of their own customizations as well. Savard estimates that MUNIS has already exposed between 300 and 400 services based on internal and customer requirements. The development team is in the process of assessing the entire system to comprehensively extend its functional capabilities as services and modernize the touch points that have historically been addressed via flat file transfers.

"SOA architecture is leaps and bounds ahead of what we were doing before," states Savard. By leveraging this approach, MUNIS was able to eliminate multiple round-trip database calls by configuring the solution to call services that process needed functions and return results. The right distributed topology in the SOA led to tremendous performance gains, especially for the company's hosted ASP solution.

The company is a VAR of Four J's. As an early access program partner and beta tester, the MUNIS team is often one of the first to be involved in Four J's releases, providing valuable input into the product. The company was also a critical member of the team that worked specifically on Web services functionality. Peter Christopher, product manager with the R&D team at MUNIS, works on special products such as joint initiatives with partners and was also a key participant in the move to Web services with Four J's.

One example that Christopher highlights is the company's current investment in a real-time decision support system that feeds into a portal environment. This offering capitalizes on Web services standards for interoperability and the flexibility of the SOA design patterns to access and process critical information from the back-end system. The open nature of the Four J's environment allows the information to be exposed as services to an add-on .NET solution that engages a larger base of customer stakeholders with critical information for reporting and decision making, especially those who do not regularly interact with the base ERP system.

MUNIS has about 100 developers using the Four J's software, with the majority using it every day as their core development tool. According to Savard, they are able to easily wrap existing code and expose Web services without having learned all the detailed XML schema required to adhere to the protocols. This is proving to be a very important and growing aspect of the MUNIS offering, fueling much of the company's ability to add new product capabilities and partner integrations. Also, he anticipates that it is only a matter of time before the local government arena embraces Web services and, due to state mandates, requires support for more specific government XML schemas.

Having an open environment is critical to the growth and long-term viability of MUNIS. Technologies wax and wane, and the ability to adjust to whatever environment customers may require is crucial. Embarking on SOA and supporting multiple computing paradigms will continue to help make the company successful, and that's what matters. According to Savard, "Sometimes the best solution isn't the biggest name."

Versaterm

Versaterm Inc., founded in 1977, is a privately held Canadian software and services company focused on providing applications and information management systems for municipal police and public safety agencies primarily in North America, including dispatch, records management, computer-aided dispatch, and wireless solutions.

Versaterm's product offerings, including its flagship product Versadex, have evolved through generations of technology, changing in both functionality and appearance. Versadex has grown from a records management system (RMS) to a strategic operational system for frontline police and safety officers. In the late 1980s, the company embraced the path of open environments, leveraging relational database systems and middleware. It chose this path to allow the system to function and advance in capabilities without being hampered by reliance on specific underlying proprietary components or needing to mandate its customers to use specific operating systems and database technology.

The Versaterm development team at one point considered moving everything off of its Informix-based system to Java; however, the team estimated that at that time, it would have taken an effort of about 10 man-years to redo the more than 1,200 screens and UI areas and 2,500 function points just to get the same level of functionality. Given the mission-critical environment of Versaterm's primary customer base, which is somewhat conservative in how it adopts and transitions to new technology, it was critical to preserve core capabilities with limited disruption.

In 1999, the company dropped its efforts to support its own "lean client" Windows GUI and adopted Four J's to address a wider range of needs in a more adaptable and cost-effective manner. The company employed Four J's Genero and its BDL with its Versadex 7 release, allowing it to add new features and expand the capabilities of its presentation layer utilizing XML. By adopting Four J's, Versaterm has been able to evolve and expand its UIs to support Microsoft Windows, Java, and Web browsers without needing to discard all of the underlying business logic. It is also taking more advantage of the product's platform versatility and Web server architecture.

According to Allan Wilson, product manager of the RMS at Versaterm, Four J's has helped the company better position itself to ultimately protect its customers' investments, integrate among its own suite of offerings, and ensure backwards compatibility between releases.

Wilson indicates that police work is traditionally very information centric and reactive in nature. It requires quick access to and sharing of a wide range of trusted and fragmented geographically dispersed sources. Thus, component integration has become an extremely important aspect of the company's product line and development strategy. Versaterm is moving in the direction of Web services and other Web-based technologies that can help merge varying content types and information sources. The company continues to develop and test different methods of utilizing and offering Web services to interface with third-party systems.

The interoperability of Web services is facilitating not only the integration of intradepartment and interdepartment transactions but also the sharing of information across multiple police force jurisdictions and justice agencies, such as border control, correctional services, and other government entities. Versaterm is helping pave the way with its Law Enforcement Information Portal (LEIP) product, an initiative based in Canada that is expanding rapidly in use and adoption. It is an operational information-sharing environment that was first designed to connect disparate RMSs. It is now targeted at providing a fully integrated justice solution (from initial arrest through prosecution and corrections), an analytical tool (to analyze crime patterns across jurisdictional boundaries), and an integrated command and control information system to help manage resources in the event of a major incident. The system is geared to help get critical and timely information to frontline officers and detectives so that they have enough information at their fingertips, even in a mobile setting, to make a decision in real time. It leverages a federated design in that all information access is controlled electronically by the contributing agency.

Versaterm has about 20 developers using the Four J's toolset and environment. Wilson indicates that Genero has made the team very productive in designing the back-end processes with the 4GL, leveraging automatic libraries, and creating Web services as simple extensions. The developers building Web services need to know only the business logic; they do not all need to be trained in the detailed protocol specifications. Wilson indicates that Versaterm is very cautious, however, in how it approaches adding anything to its system environment because it needs to ensure that it remains highly reliable, secure, scalable, and high performing. For example, if the design calls on any new system libraries, Versaterm needs to consider what the cost and benefit might be to processing flows. It also must make sure that any supporting software releases are production ready; thus far, Versaterm has been pleased with Four J's abilities in this regard.

When the basic Web services protocols were still in the early stages of formation, it was more difficult for the company to embark on a path when there was potential volatility in what specific syntax was being supported. Now that the basic specifications have stabilized, combined with the government moving to a standard justice model as well as token-based user authentication system, Wilson believes that even more use of Web services will unfold.

Versaterm is also retooling its dispatching back-end system to integrate with a new .NET front end, pointing to the flexibility of the Four J's Genero tool to facilitate such an open environment. In a way, Four J's provides the company with some level of insurance, allowing it to react to any changes in the software industry. Versaterm believes that its commitment to Four J's is really based on a two-way partnership in the way that it gets value back from the relationship. When Versaterm needed certain requirements, Four J's listened to its needs and provided an early release of the software to fulfill those capabilities.

Versaterm is able to address customers with departments that span from 25 to over 5,000 users, and the company continues to experience high growth rates as it expands geographically and through channel distribution. The company attributes much of this success to its ability to supply a modern open system environment and leverage flexible tools to address the unique dynamics of its constituency. This ability has also allowed the firm to build in functionality that takes advantage of the latest in browser-based presentation and wireless technology.

When asked about any plans to change course, Versaterm indicates that it is very satisfied that Four J's stays either ahead of or right in line with its needs. The company can focus on thinking about the business first. As Wilson states, "Why take any risks? It's hard to argue with success."

CHALLENGES AND OPPORTUNITIES FOR FOUR J'S

In today's highly competitive software arena, Four J's must raise market visibility for its company and offerings, dispel any remaining notions that its technology is dedicated to Informix-only environments, and further educate its clients and prospects regarding methods of utilizing open systems, including SOA constructs and Web services.

The company is focusing its R&D efforts based on customer feedback. Conversations with these reference customers indicate that the demand certainly exists for Four J's to press forward in its endeavors to support Web services standards and SOA. It will be important for the company to stay ahead of the curve in this regard. SOA is a very collaborative and shared environment.

IDC believes Four J's could work to enhance its overall IDE capabilities and help organizations interface more readily with complementary service life-cycle technologies (such as adding the ability to integrate and search service registries, publish to target registries, add plug-ins to other frameworks, and more). Another element of architectural and development trends to keep an eye on is the business process orchestration and business rules arena. Being able to integrate with, if not offer, these kinds of technologies would potentially add significant benefits.

Most importantly, raising awareness of business stakeholders is critical for its IT fans, as mindshare and name recognition are always important considerations in the decision-making process. Leveraging the volume and name recognition of its base of customers and partners will go a long way in spreading the word. As the application market matures, and increasing levels of importance are placed on providing domain-specific functionality, midmarket and industry-specific vendors, some of which compose Four J's customer and distribution bases, should glean more attention as well.

CONCLUSION

One of the key themes that emerged from the Four J's case studies is that both companies interviewed are concentrating on providing an integrated and rich experience for their customers, allowing for the choice of interaction mode and system infrastructure foundations. They also understand that although change is needed to address business needs, it is difficult for their customers to readily orient their environments to whole new processes and technologies. Creating a path of gradual innovation is the road map for many to best pursue.

Most organizations cannot readily rip and replace their existing systems due to a large number of factors such as resource constraints and risk mitigation. Enterprises that had originally thought the only ways to modernize their legacy systems would require intensive recoding efforts are beginning to breathe life into these applications by enabling them to function as equal participants in an SOA environment. Although such legacy systems may eventually be displaced over the long term, ensuring that they are used to full capacity in the interim and that the processes and systems that call upon them are well understood before attempting massive overhauls is important. It really is about seeing the entire computing and business landscape as a whole versus perpetuating the age-old problem of addressing particular solutions in technology silos.

Four J's is in a good position to help organizations extend solutions across a variety of domains with little to no disruption. In today's fast-moving business environments, a RAD tool that can help absorb the complexity of the finer-level intricacies of configuring to specific systems should be a technology in demand. By creating a flexible environment that addresses multiple system configurations, Four J's is enabling its customers to choose, add, and replace elements of their infrastructures without changes to core application logic. These applications can readily become service enabled, especially if structured design tenets have been employed, and interface with an even larger ecosystem of enterprise processes and technologies.

Technology, however, goes just so far. For application architecture and development, it really is all about implementation. Learning and fostering best practices, while taking advantage of tools that reinforce strong design tenets, are the most critical factors in the end. When they are combined with deep knowledge of the business domain, the result is empowerment.

Copyright Notice