Next Generation Intelligent, Mobile, Widely Distributed Applications:
Solving Tomorrow's Software Development Challenges Today

By Bob DeAnna, Chief Technology Officer, Recursion Software

We are on the cusp of the next giant step in software applications. It is a new frontier, which is there for the bold of mind to embrace. This new caliber of applications will be hugely beneficial to mankind, the quality of all of our lives, and the safety and security of our nation. These applications will be pervasive and impact every aspect of our lives…how we work, learn, communicate, receive medical care, travel, shop, and play.

Today's development environment demands next generation tools that will empower corporations and organizations to build the killer applications of the future. These applications will be intelligent, mobile and able to interoperate with numerous platforms and protocols These applications will be self-learning, widely distributed and dynamic, and run on multiple wireless networks, as well as embedded or wireless (edge) devices while maintaining interoperability with many different enterprise systems.

Let's review the current environment in commercial, government and consumer arenas today, and we will see why the stage is set for these next-generation applications and the software platform that they will require.

Today's Environment of Expectations

The demand on technology infrastructures, networks and applications has reached a critical mass where we are increasingly more dependent on our application infrastructures in order to conduct our daily business activities. As the nature of work becomes increasingly geographically distributed and edge computing prospers, the bar for performance is raised higher and higher. Evidence of this fact abounds:

Devices:
Estimates have the number of handheld and embedded devices entering the market exceeding one billion per year. These devices include PDAs, smart-phones, portable entertainment devices, and other embedded/wireless devices in our homes, automobiles and workplaces.

Data:
The amount of information and data being passed over wireless and wired networks is exploding due to the enormous increase in "edge" devices. These devices have increasing storage capacities and produce and aggregate an ever-expanding amount of information. Professionals and consumers also want to share information, often dynamically. with peers and peer groups. They do not want to be restricted to their desktop or their enterprise networks and databases. Another obvious reason for this information explosion is the exponential growth of the Internet and the increasingly dynamic terabytes of information contained on it, streamed from it, and the volume of devices accessing it.

Networks:
Today's networks, particularly wireless networks, are unreliable and/or not powerful enough to adequately handle the loads that are on them, and will be put on them in the near future. There are also many existing and emerging wireless protocol standards that are geared for particular uses and geographic areas, ranging from Telco networks, Wide Area Networks, Local Area Networks and Personal Area Networks.

Application Software Stacks:
Today's applications are still predominantly client-server based, with rigid business intelligence. They are too reliant on networks and/or are too centralized. The vast majority of today's applications are also not adaptive or self-learning.

Corporate Consolidation:
Corporate mergers and acquisitions have wreaked havoc on traditional information technology systems. The combined IT divisions must integrate their applications and software, but nothing is consistent. Systems are standardized on a variety of different enterprise software stacks, servers and mobile devices. Some people call this "the hairball."

Next Generation Applications
To move forward, the next generation of applications will be different. Development will occur on a platform that allows for decentralized, intelligent, mobile, high-performance, transaction-based applications. The platform will have to be seamlessly interoperable with the hardware and software technologies of today, will pave the way for next generation solutions for commercial, government and consumer development.

These applications will provide quantum leaps in real-time intelligence capabilities and coordination among widely distributed groups, as well as offer greatly increased productivity and cost-savings.

In this paradigm, data will be gathered on edge devices, rapidly turned into actionable knowledge, and be distributed efficiently to ad-hoc, possibly dynamically assembled groups of interested participants, devices and enterprise systems.

Mobile software components (agents) residing on these edge devices will continuously learn from the data they receive and interpret. This may cause them to respond differently and possibly assemble and communicate with an ever-changing audience, forming a distributed knowledge-sharing network.

But to make this vision a reality, these killer applications will need a next-generation software framework that will meet a long list of requirements to seamlessly operate in the complex, widely distributed environment we find today.

Next Generation Attributes
We believe that a software platform must meet certain requirements to provide the highway upon which tomorrow's applications will operate. Specifically, the next generation platform must:

I. Increase Productivity and Extensibility via seamless .NET, Java and Legacy interoperability
Next-generation applications must seamlessly operate in .NET, Java and any combination thereof, in a high performance manner. Creating remote mobile software components, or agents, in either language will leverage pre-existing code and increase programmer productivity. Also, applications must address the need for seamless integration with external systems and applications.

II. Increase Network Survivability and Mobility
Next-generation applications must be able to process data at the source to minimize network traffic and handle unreliable and/or limited network connections and adjust to hardware failures or CPU load. The software components or agents running on edge devices must be able to dynamically reconfigure themselves to use a communication protocol that best matches the capabilities of their current network

connection. They also must be able to move to nearby nodes if device failure is imminent, due to battery failure, disk failure, etc.

III. Provide a Single Unified Platform for .NET and Java
Next-generation applications cannot be limited to a single development environment and programming language. Furthermore, the exact same API would be provided to both .NET and Java developers. This would greatly increase programmer productivity and allow developers from both "camps" to easily work together and share software.

IV. Decentralized Messaging
Next-generation applications must be able to communicate to groups of devices/systems without the need of a centralized messaging server. It must support multiple levels of granularity of the intended audience for a given message. It must also support messaging over standard centralized messaging servers for integration with enterprise and legacy systems. More specifically, there must be seamless integration with Microsoft's Message Queue (MSMQ), Java's Message Server (JMS), Object Management Group (OMG)'s Data Description Service (DDS) and CORBA Notification Service.

V. Mobile SOA Architecture
Next-generation applications must provide a Service Oriented Architecture (SOA). SOA creates an environment where loosely coupled mobile agents can communicate with any Web Service client and vice versa. These Services must be accessible via centralized Web Service Container such as Microsoft's Internet Information Services (IIS) and those provided in the Java world, but also must be accessible in a decentralized fashion directly to agents, exposed as web services, that are running on edge and wireless devices. The location of intelligent mobile agents and the Mobile Web Services they expose, must be irrelevant to the web service client. Finally, all agents must be accessible by a Service Description, in a Yellow-pages directory, ideally one that is Universal Description, Discovery, and Integration (UDDI) compliant.

VI. Transactional At the Edge
Next-generation applications must extend transactions from the enterprise to also include edge devices to allow for distributed, but coordinated, tasks among peers, peer groups and the enterprise. To do so, support must be provided to allow for intelligent agents running on the edge to participate in enterprise transaction managers such as those provided in .NET's IIS, Java's Transaction Service and OMG's Object Transaction Service.

VII. Wireless/Embedded Device Support
Next-generation applications must be able to run, in some form, on all devices. To do so, the next-generation platform must be pervasive and supported on these same devices. Intelligent mobile agents also need to be able to run in the popular embedded software stacks such as Microsoft's Compact Framework, Java's Micro Edition and OSGI Containers on a wide range of embedded operating systems such as Windows Mobile, Symbian and Embedded Linux.

VIII. AI-Rules Integration
Next-generation applications must contain intelligent agents that gather data, respond quickly based on this data as it changes, produce and distribute knowledge and possibly initiate other agent activities. The underlying rules engine must be easy to use, provide very high performance against potentially large rule sets, and must be available in .NET and Java.

IX. Security
Next-generation applications must provide a high level of security to ensure privacy, and protection from rogue/viral clients and mobile agents. This will involve security agents and agent managers that provide capabilities above and beyond the current encryption, authentication and authorization that are currently employed in today's centralized client server applications.

X. Database Integration
Next-generation applications must provide a simple way to access databases, regardless of the type of database, be it relational, object, xml, and a multi-user enterprise database or a single-user embedded type. This has the obvious benefit of increased programmer productivity and results in consistent database access software.

XI. Extensive Management, Monitoring and Self-Healing
Next-generation applications will run on many nodes and devices, and thus the ability to monitor and manage the agents on the devices is imperative. These agents must be able to monitor themselves, and the nodes upon which they run, while providing a self-healing, preventive capability to preempt agent failure.

XII. Extend and Simplify .NET, J2EE and CORBA
Next-generation applications must easily and seamlessly integrate and be interoperable with the popular enterprise software stacks. There should be one distributed platform that unites these stacks. Microsoft technologies, including .NET have been *successfully* developed with ease-of-use in mind. One could argue that the same can't be said of Java Enterprise (J2EE) and CORBA. The next-generation software platform will provide an easy-to-use, high performance way to allow intelligent agents to communicate regardless of which enterprise software stack they are currently resident on, and will have the freedom to move between them.

Next Generation Benefits
A software platform that meets these requirements brings revolutionary capabilities and ease of use to today's application architects. It gives them maximum flexibility to freely develop dynamic, intelligent and decentralized applications in .NET and Java, on the devices and servers they need to target. Additionally, the extensive protocol support -- in combination with its multi-language support and SOA -- allow the intelligent mobile agents in these applications to access systems written in any language. Additionally, these applications can be accessed by systems written in any language, in a variety of ways, using various messaging techniques, be they centralized or decentralized.

By leveraging the AI-rules integration and the pervasive mobile agents residing on any or all targeted devices and servers, organizations could create applications that provide real-time intelligence, coordination and capabilities not found today.

There is a great deal of momentum moving the enterprise toward the .NET architecture, but there are a plethora of Java, CORBA and Mainframe systems within many of these same organizations.

A very similar situation exists in the device world. Windows Mobile is gaining speed and acceptance amongst device manufacturers, but there will always exist devices running Symbian, Embedded Linux and even PalmOS.

This unifying platform will allow next-generation applications to seamlessly exist in this diverse software environment and the increasingly diverse wired and wireless networks upon which they are running.

This platform provides the capability to create a new set of applications that are distributed knowledge and problem-solving networks with dynamic and ad-hoc participants using different devices and enterprise software systems.

It is not hard to see where this could be of great benefit in the domains of disaster recovery, terrorist response, and homeland security to name only a select few applicable areas within the Government. In the commercial arena, the energy, transportation, insurance, education, telecom, healthcare and finance industries are obvious verticals with vast new opportunities for such a platform. Indeed there is not a vertical in the commercial, government or consumer arena that could not create revolutionary new capabilities with a platform of this type to leverage and utilize.

Just Around the Bend

This next generation platform might be much closer to reality than you think. So let your creative mind take charge and start envisioning the revolutionary, world-changing applications that can be developed with these new capabilities. And it will be extremely profitable to those who create them.

It is indeed extremely exciting, inspiring and motivating to think about and work toward this unifying platform and the killer applications it will enable.

About the author

Bob DeAnna, CTO of Recursion Software, has 22 years of experience in software architecture, and engineering. Bob's expertise is in distributed and wireless application frameworks such as J2EE, CORBA, ATMI and J2ME. He has architected and developed applications and middleware in Java, C/C++, COBOL, PLI, and Assembler, hosted on operating systems ranging from MVS to Unix, Linux, Windows to Symbian and other embedded OSs. Bob also is a co-author of a patent and commercial implementation thereof, for servers on embedded and portable devices such as PDA's and smart-phones.

Bob received a BS in Mechanical Engineering from Rutgers University and a continuing education degree in C/C++ and Unix Programming from New York University.