Mono and the Java vs .NET Balance of Power

Stephen O'Grady

While the Mono open source project did not generate that much noise amidst the fanfare of Novell's two Linux-related acquisitions of SuSE and Ximian, it is now set to take the application development center stage. Mono's ambitious goal is to reverse engineer and replicate portions of Microsoft's .NET framework to enable cross-platform, open source development and deployment projects. On June 30th, Novell released Version 1.0 of the platform.

Mono is a project that potentially unites Microsoft development capability with Linux deployment flexibility, and is being enthusiastically adopted by Linux developers. Mono is also, however, a project that has been accused of creating a schism between the Java and Linux communities, which some contend will ultimately benefit Linux's arch-enemy, Microsoft. Mono has become one of the most important open source software development projects in the application development space. This assertion of preeminent importance is a big claim, but justifiable. The reasoning behind it is simple: Mono has the potential to upset the delicate balance of power between the two major application deployment camps, Java and .NET, because it potentially decouples .NET development from Windows deployment.

> "Mono has become one of the most important open source software development projects in the application development space."

What is Mono?

Mono kicked off in 2001 as a an attempt to replicate Microsoft functionality on non-Windows platforms. Like Microsoft .NET[1] it consists of three main pieces: a C# compiler, a Common Language Infrastructure (CLI) runtime engine, and a set of class libraries (including both .NET and Mono-specific components).

The portions that mimic .NET functionality are derived from two key technologies that Microsoft submitted to the European Computer Manufacturers Association (ECMA) standards body in October of 2000 - the C# language and the CLR runtime. Since its inception in 2001, Mono has evolved to support APIs compatible with the Microsoft .Net Framework 1.1, including support for ASP.Net and ADO.Net. Practically speaking this means there should be good platform compatibility in one direction going from .NET to Mono, with more issues going the other way due to the delivery of unique Mono API libraries.

Mono represents a port of the C# language and a set of .NET-like libraries that should allow .NET developers to design and build components suitable for deployment to multiple platforms that support the Mono runtime, including Linux, Windows and OS X. Mono's own integrated development environment (IDE) obviously suffers in comparison to Visual Studio .NET, but the project is still in its infancy. Developers can also use Visual Studio .NET, as Mono is compatible with ASP.NET or .NET console executables output by Microsoft's IDE, with the notable exception of Windows.Forms based GUIs which are not yet supported. The Mono team is currently working on plugins for Visual Studio that would allow developers to test a Mono application on Windows prior to deployment, as well as migrate Visual Studio projects to makefiles or MonoDevelop projects. Going the other direction, MonoDevelop has the ability to import Visual Studio .NET projects directly.

[1] Microsoft may have backpedalled on its .NET branding – but in the world of enterprise architecture .NET is generally understood as a reference to both the managed code model, and the platform defined by the Microsoft CLI, the associated set of APIs and class libraries and the Visual Studio. NET development environment.

The current MonoDevelop IDE is itself a port of AlphaSierraPapa's open source SharpDevelop IDE. But porting MonoDevelop to the open source Eclipse platform – a plugin is reportedly underway - would potentially touch more developers and so would be likely to drive more rapid adoption of Mono. While Eclipse is a toolset primarily oriented towards Java development, it's less wedded to Java than say, Sun's netbeans.

Mono and the Community

While Mono began as an attempt to bring new functionality – based on the .NET platform – to the open source community, the community has had mixed reactions. To say that Mono is a controversial subject is to put it mildly. Many in the community are opposed to Mono either because it is aligned with Microsoft – a longtime community enemy – or because of concerns around potential licensing and patent issues, or both. Open source developers associated with Red Hat have been among the most vocal in their criticisms.

"While Mono began as an attempt to bring new functionality – based on the .NET platform – to the open source community, the community has had mixed reactions."

But other open source advocates, more pragmatically focused on the perceived productivity benefits of .NET application development, have embraced the Mono concept wholeheartedly. While the technology itself is relatively newly minted, we're already seeing a surprising amount of Mono pilot and skunkworks projects underway. From Novell centric-efforts such as iFolder, Dashboard and Beagle to commercial offerings such as OpenLink's Virtuoso – a middleware component for database connectivity - to underground projects such as Jon Lech Johansen's (better known in some circles as DVD Jon) FairKeys – a simple application that allows users of Apple's iTunes to fetch their DRM keys from Apple's servers, Mono is a community with some momentum. HP and Intel saw enough promise in Mono two years ago to invest in the technology.[2]

Unsurprisingly, Novell itself has adopted Mono as its de facto development environment, and much of its new development is targeting Mono as the development platform rather than Java, as might be expected of a firm that's reaffirmed itself as a Microsoft alternative. Just as Novell has made a commitment to Linux as its future for the operating system foundation, so too has it committed to Mono for its development. Rather than develop important new applications like iFolder –Novell's answer to Longhorn's WinFS – in C, C++, or Java, the decision was made to develop the application in Mono. Since the Linux desktop is an important area of investment for Novell – important enough to justify the acquisition of Ximian a year ago – such decisions are not made lightly.

Mainsoft has made interesting use of Mono in its development of Visual MainWin, a tool which allows a user to build an application using Microsoft ASP.NET before deploying to a Java 2 Enterprise Edition (J2EE) server. Java components can be accessed directly by Visual Studio .NET before deploying the subsequent application to Apache Tomcat or any J2EE server. Mainsoft has been a strong supporter to Mono, donating a Java implementation of the Microsoft.VisualBasic library that is now being ported to C#.

To further emphasize Mono's cross platform credentials it is worth mentioning a couple of other current projects. Just as is the case with Linux, the platform is making its way into some interesting real estate: ports to IBM's mainframe S390 operating environment and Sun SPARC arrived with Version 1, while AMD's 64-bit extensions are underway.

[2] 1.28.02, "Intel, HP back Mono effort for open .Net," ITWorld.com
[3] 5.19.2004, Seth Nickell, "Why Mono is Currently an Unacceptable Risk."
[4] 5.20.2004, Nat Friedman, http://www.nat.org/2004/may
[5] 12.2001, Dare Obasanjo, "Using the ECMA Standards: An Interview with Miguel de Icaza," MSDN
[6] 7.15.2004, Martin Lamonica, "More than an open-source curiosity," News.com
[7] 6.26.2003, James Governor with Stephen O'Grady, "Evolution and Extinction: The Application Server Market in 2003 and Beyond," RedMonk
[8] 12.10.2002, Thor Olavsrud, "META: Microsoft to Offer Server Apps on Linux?," internetnews.com

Mono: Not Missing in Action

With Mono, Germany is yet again at the forefront of open source adoption. Just as it has with Linux, the City of Munich is playing a key role in being an early adopter for Mono. Voelcker Informatik, a German software and consulting firm, is assisting Munich's department of education with the migration of 14,000 Windows desktops and some 400 NT servers to Linux. Voelcker is also porting its IT Service Management application, called Active Entry, from its current Microsoft COM-based architecture to Mono; the package offers identity management, help desk, asset management, and provisioning services and is scheduled for release at next year's Cebit show. If Voelker pulls it off and builds a strong Mono-based service and asset management system, it would make them a potential Novell acquisition.

Licensing and Patents

So what about patent issues? There are two main areas of concern. First, technologies contributed to the ECMA by Microsoft are not donated to the community under GPL-like terms, but offered up under nebulous – and to many, problematic - "Reasonable and Non-Discriminatory (RAND)" conditions. As Seth Nickell, a GNOME contributor, puts it, "RAND + royalty free is very different from a patent grant."[3] While the nuances of licensing law are best left to the lawyers, the crux of the issue for critics is that by licensing these technologies – even under theoretically reasonable and non-discriminatory terms – the open source community becomes beholden to a commercial software organization that has been a highly aggressive critic of open source software methods, technologies and approaches. Further, what's viewed as non-discriminatory to Microsoft may not be similarly regarded by the open source community.

"To date, Microsoft has not only failed to slap lawsuits on Novell, but has seemingly encouraged Mono, if in a subdued fashion."

Second, beyond the ECMA pieces are the libraries that are being replicated by the Mono project. Those lobbying against Mono contend that there is a possibility that there may be undiscovered patent infringements in the project, which could translate into legal difficulties down the line as they are discovered. Novell's Nat Friedman, however, contends that Mono has reviewed "hundreds of patents and know of none that threaten Mono."[4]

Without getting into the legal risks or lack thereof within the Mono project, it's interesting to observe the behavior of the perceived and potential antagonist in this unplayed- out legal drama, Microsoft. To date, Microsoft has not only failed to slap lawsuits on Novell, but has seemingly encouraged Mono, if in a subdued fashion. Novell's Miguel de Icaza, the father of Mono, has been interviewed for a feature on Microsoft's developer site[5], MSDN, and recent reports had Microsoft demoing Mono technologies for a Linux crowd.

It is also important to understand that MS's patent portfolio could be used to stymie all sorts of interesting developments, given that is has been awarded one patent for using the human body as a conductor for data processing, and one for single click execution on handhelds, and so on. Mono is to some extent no more or less likely to escape Microsoft patent lawyers than any other technology.

.NET, Not Java

Why Microsoft and .NET rather than Java is the question implicit in most of the conversations about Mono. Some in the industry would use Mono as an indictment of the Java Community Process (JCP), believing that a key driver behind the project was the lack of fast paced equivalent innovation within the JCP. While many have complained about the JCP, however, nobody has suggested a better model that withstands any scrutiny. The point is moot, anyhow, if we're to judge from de Icaza's comments. It is apparent that a break from Java was much more fundamental than a simple procedural frustration – Ximian, and de Icaza specifically, disagreed with Java at a basic level. -

Instead, Mono is evidently the result of a conclusion that Java is more suited to high end tasks than to more mundane, day-to-day tasks that occupy most developers. As de Icaza put it,

> “The problem with J2EE really is that it became very, very academic and the complexity of all these perfectly designed systems in schools does not necessarily map when you have deadlines and all kinds of other things.”[6]

This perception is not new; RedMonk has expressed similar sentiments regarding the commercial application server market;

> “Not every organization is a Charles Schwab or an eBay, but BEA and IBM tended to compete as if that were the norm, not the exception.”[7]

Too often, vendors pushed, and still push, complex J2EE runtimes when simple, “good enough” servlet based applications running in Tomcat would have sufficed. The net result is that J2EE, and therefore Java development generally, is perceived as too complex.

To its credit, Java has not been static in this area; we're seeing an increasing focus from the JCP on attracting new, more junior developers through standards like Java Server Faces. These standards have given rise to new tools such as Sun's Java Studio Creator and IBM's WebSphere Studio Site Developer, which are aimed at enhancing Java productivity in lightweight forms-based development.

“The net result is that J2EE, and therefore Java development generally, is perceived as too complex.”

But as far as Mono is concerned, Java's evolution is too little, too late. The choice for Miguel de Icaza and Mono was made in 2001, and at that time Java wasn’t meeting their expectations for developer productivity.

Mono's Implications

Given the absence of hard evidence of project-threatening legal issues, not to mention the fact that Novell, a $2B+ enterprise software concern, is betting a large part of its product development on the technology, we expect momentum for Mono to build over the coming months. What are the implications for enterprise software concerns? We see them as follows:

Java

- **Developer Drain**: While Java and .NET will continue to coexist as enterprise standards for the foreseeable future, Mono allows developers to explore the .NET platform without sacrificing the cross-platform compatibility of Java. This, more than any other factor, has the potential to impact the Java community. During a debate on open sourcing Java that RedMonk participated in at the JavaOne Conference, Tim O'Reilly implied that Java is being pressured by Mono, saying “There is an elephant in the room--the fact that on Linux, the Mono project is implementing .Net, not Java. Microsoft does have a foot in the door."

- **ISV Recruitment**: Existing industry ISV platform decisions and relationships of course won't immediately be affected, as most are longstanding and unlikely to be swayed by a relatively new project like Mono. We are not claiming an immediate broad realignment around .NET and Mono. But in ISV recruitment, or where new, strategic platform decisions are made, Mono has the potential to impact the decision making process because for the first time Visual Studio .NET class development capabilities will no longer be restricted to Windows systems.

Microsoft

- **Broadened Horizons**: Mono provides Microsoft with a low profile entry onto non-Microsoft platforms. While many have postulated that Microsoft's reaction to Mono would be conflicted at best, we believe that Redmond is likely to be – privately, at least – ecstatic about the project. Through Mono they can obtain cross-platform exposure for their platform - the same cross-platform opportunity that led some to believe Microsoft would support Linux with its future software offerings[8] - without dropping its virulently anti-Linux stance. This is a big fear in the Java camps; that is—that Microsoft is effectively getting a *cross platform deployment strategy by proxy*. Hence Mono might play a role somewhat similar to the Eclipse Project, which insulates smaller vendors from IBM,but at the same time gives IBM even more market power as a major contributor to the project.

- **C# Propagation:** Considered to be a state of the art language for software development, many universities are already building computer science courses around c# and the .NET framework. Examples include three universities in Berlin and one in Dresden. Without ECMA standardization and the potential for cross platform develoment, many universities, especially those in Europe, would have avoided using a Microsoft reference architecture. Mono then indirectly accelerates the building of the NET skills base.

.NET Vendors

- **Platform Choice**: ISVs and .NET ecosystem vendors can begin casting a wider net; by porting applications to Mono, vendors can potentially target entirely new platforms such as Linux and OS X for their applications. Voelcker Informatik is a great example of this kind of thinking in action.

- **Attract New Developers**: Mono gives free and open source developers a less conflicted – if not issue free – opportunity to explore the .NET platform and its capabilities versus the more traditional approach of Java.

“It is no accident that Mono was a common topic during the panel debate on open sourcing Java at JavaOne – it is looming as the biggest threat that Java has faced in some time.”

RedMonk Take

Love it or hate it, Mono is evidently here to stay. Nascent though it may be, its implications and impact for the developer community, platform landscape and Java vs .NET competition will be widespread and profound. It is no accident that Mono was a common topic during the panel debate on open sourcing Java at JavaOne – it is looming as the biggest threat that Java has faced in some time. Incorporating the advantages of .NET without its single platform limitation, Mono is emerging as a significant, disruptive force. Vendors in the Java camp should be evaluating it as a competitor for developer numbers and platform dollars, while ISVs in the .NET camp should consider how it may affect operating system decisions and open new doors for them.

What of enterprises? For either the heterogeneous enterprise with a mix of both Java and .NET or the pure .NET organization, Mono certainly warrants a closer look, even thought its very early days for the technology. Both Java and .NET will continue to have their place in enterprise IT deployments, but Mono offers the opportunity to simultaneously embrace technologies such as Linux and .NET, previously considered to be oil and water. For an organization that sees advantages in both development worlds, but would prefer to standardize on Linux at the operating system level – whether for cost reasons, security concerns, or simplified management – Mono may prove to be an attractive option. There may also be a market for frustrated Java development organizations that would like to embrace .NET, but are reluctant to commit to Windows as a platform of choice.

We recommend, of course, that any organization strongly considering Mono as a strategic platform invest in due diligence work, in order to mitigate and analyze the potential licensing and patent risks discussed above, but Novell is certainly eating its own dog food here, which should be some assurance. Patent and licensing dangers would seem to represent more of a risk if Mono wasn't indirectly aligned with Microsoft's own best interests.

Mono's future will be interesting in any event, but given the amount of influence it has on the two dominant application development communities it's no stretch to say that Mono is shaping up to be one of the most important open source initiatives in the industry. Enterprises and vendors on both sides of the Java/.NET divide should be determining what impact Mono might have on their strategic direction. In a world where choice and interoperability are always paramount concerns, Mono just may open some new doors for CIOs, not to mention a new front in the ongoing Java versus .NET war.

> “Enterprises and vendors on both sides of the Java/.NET divide should be determining what impact Mono might have on their strategic direction.”

About the Creative Commons License

About RedMonk

RedMonk is a research and advisory services firm that assists enterprises, vendors, systems integrators and corporate finance analysts in the decision making process around today's enterprise software stacks. We cover the industry by looking at integrated software stacks, focusing on business and operational context rather than speeds and feeds and feature tick-lists.

Founded by James Governor and Stephen O'Grady, and headquartered in Bath, Maine, RedMonk is on the web at www.redmonk.com. If you would like to discuss this report email sogrady@redmonk.com.