From: **Bill Gates**
Sent: Mon, 2/15/1999 11:02 AM
To: Eric Rudder
Subject: FW Microsoft: (ZDNET) Mulls Java Competitor

Jim's point of view.
-----Original Message-----
From: Jim Allchin (Exchange)
Sent: Monday, February 15, 1999 10:57 AM
To: Bill Gates
Cc: Paul Maritz
Subject: FW: Microsoft: (ZDNET) Mulls Java Competitor

I know you are a strong believer in cross-platform. You have asked me what I think about it. Although I may not be directly involved in this area after the reorg I wanted to share my thoughts with you. The press below also discusses the cross platform issue.

I see two issues surrounding the push for cross platform. The first is Java/Jini itself and the second is EJB. I see these issues as potentially separate. I could be wrong, but I believe our biggest problem is really java/jini.

1. So what do I think we should do about Java/Jini?

- Getting momentum back in our languages/runtime is critical. This also translates into additional momentum for the platform. So, I am supporter of creating a new language (and cross language IL) which will attract developers back to us. Java is still somewhat foreign to C developers and certainly foreign to VB developers. We could do something really great for developers here.

- To overcome inertia, it will take something massive. I propose we open source at least one part of the IL interpretation (remember there isn't a classical interpreter in the current design). If we did have a classical interpreter, it should be open sourced. (Perhaps, we should create such an interpreter.) In order for the language to take hold it will have to be available on virtually all machines (think availability of the old Pcode interpreter). We will have to sign up some others (e.g., HP) to ensure that it is ported to seed it, but after we kick start it, open sourcing it may push it the rest of the way. We should take the language itself and offer them without strings to ANSI/ISO, etc.

- To offer an alternative to Jini we need to do two things: finish/push our resource discovery protocol work and get COOL/IL promoted widely as free as water. I think our resource discovery approach is stronger than the Jini approach, but it is not enough. In order to ensure that our new runtime is available on small devices we need to show SONY and others in the consumer electronics space that there are advantages of COOL -- and in particular the IL This is a little strange since this path suggests we won't receive any royalty on these devices. Unfortunately, I don't see any way out of this. Java/Jini is free to these companies today. Although our technology is better, given their price sensitivity on these devices, it is hard for me seeing them paying anything. We do have an opportunity with Windows CE by doing a great job by including an awesome implementation of the IL. This could save the consumer electronics companies time in software design, etc. In order to win in this area though the Windows CE team must renew their efforts to go smaller/simpler. This makes sense given the Palm threat anyway. But, whether we are able to sell a product or not, the strategy I outline still makes sense because we need to ensure that the "PC" fills the key controller role in the home.

Plaintiff's Exhibit
6555
Comes V. Microsoft

Somewhat separate from the above, we need a coherent, simplified API set: think "class library" for Windows. (We sort of have this today with MFC, Vbrun, etc. which is what most developers use.) However, the issue today is that this class library environment doesn't integrate HTML in such a way that the value of having client side code/data is recognized. We need to design a solution so that reach is possible for simple devices, but powerful clients receive deeper benefits transparently. This is the area where although there has been good thinking, there isn't a technical design yet.

I have doubts about the services area of COM+. This isn't to say that xactions are good and the implementation isn't good. I worry about the deeper architectural framework. I believe that a fairly complete rework is required to start from a more fundamental message passing underpinnings. Some people are thinking about this, and perhaps because I have been away I haven't seen any progress here. I do believe david understands this, but there is a lot of work from where we are today. I am concerned we look at only individual pieces and we never see a total stable design. At least that's my opinion. I sometimes feel each piece is changing a lot. It is still all called COM+, but what COM+ is (again talking about the services) is changing a quite a bit. It would be great to see a total design look at this whole space. Again, because I have been gone maybe things are better than what I fear here

We could call this new Windows API something like Windows NG -- for windows next generation I have recommended something like this before.

2. So what do I think we should do about EJB?

There are two aspects of EJB: the API specs and implementations. Having the API specs for the class library as discussed above, is something that may make sense. This is a simple decision. If we are creating a virtual OS. then promoting these APIs everywhere makes sense. I am not hardcore one way or another on this. In general I'm more of a supporter than a nay sayer. It all depends on how powerful our story is above on Windows. If we have the coolest technology on Windows, then this is less important. If we have so/so technology, then it becomes critical in a hurry. I am still a believer in doing it great on Windows is the answer and just doing what customers ask for on other platforms second is the right strategy.

Creating an implementation of the entire class library on other OS systems is something we could do. However, given the current code base, I think porting MTS, kemel parts of IIS, etc. will be a lot of work. There are NT specific aspects of this code today which would have to be removed. I worry about our focus doing this vs. focusing on getting a new design for my point 1 above done. Provided we charge for this layer I don't have a business reason not to do it. My concern is about our focus and the chances of deep success quickly.

I see no reason to port SQL/MSDE technology. I'm sure an argument can be made for doing this, but my view is that this is work that will undermine NT Server and it will take away people who could do more advanced things on Windows.

These thoughts were just off the top of my head. If you want to discuss any of these thoughts more, just let me know.

jim

-----Original Message-----

From: Library News Service
Sent: Monday, February 15, 1999 8:35 AM
To: Daily Newswire Subscribers
Subject: Microsoft: (ZDNET) Mulls Java Competitor

Microsoft Mulls Java Competitor

But Will 'Cool' Be Overrun by an Internal Faction Fight? Stay Tuned

by Mary Jo Foley, Antone Gonsalves and Scott Berinato, Sm@rt Reseller
ZDNET
02/15/99

Microsoft Corp. has begun briefing developers on an alternative to Java that would be free of Sun Microsystems Inc. technology and license obligations.

The Microsoft (Nasdaq:MSFT) software, which could take the form of a new language, could be similar to Java in its object-oriented design that has attracted so many developers to Sun's programming language, sources said.

Still unknown, however, is whether it will include Java's cross-platform capabilities or be tied strictly to Windows.

The Redmond, Wash., company is using the code name "Cool" to refer to the Java alternative, according to developers and Microsoft insiders.

Cool a Java Competitor

"Cool is Microsoft's Java language competitor," said a developer close to Microsoft, who spoke on the condition of anonymity. "We are aware of it and have been watching it."

Microsoft officials acknowledged that the company has discussed the concept of a new language but insist that it's not a major focus.

"Yes, we are looking at lots of new things," said Michael Risse, product manager for Microsoft application development tools. "Where is that in the overall context? It's in the 10 percent exploratory case. Nobody is writing any code to any new language in this company today and in the foreseeable future."

Beyond the Whiteboard?

Outside of Microsoft, however, several partners claim Cool is more than just an "idea on a whiteboard," as Risse characterized it.

Rather, they say, it will epitomize Microsoft's belief that users want a Java-like product more for its productive development environment than for its cross-platform support.

"We're not just talking about C++ extensions. Microsoft needs something radically new, and that's what Cool is supposed to be," said a Windows developer who has worked closely with Microsoft and requested anonymity.

A third developer and Microsoft partner added this: "Everyone has a different idea about what Cool is. Some people are saying Cool will be based on the technology Microsoft acquired from Colusa [Software Inc.] a few years ago. [Colusa was] building a run-time language like Visual Basic. Microsoft has offered a proposition for a new language that looks a lot like Java but is missing some of the key features of C++."

Fighting Over Cool

Sources close to the company say factions within Microsoft are fighting over the language strategy. While some executives are pushing for a completely new language, others favor beefing up C++ to make it better than Java, sources said.

In addition to using the code name Cool to refer to an entirely new language, Microsoft also has used Cool to refer to extensions to its Visual C++ language.

News of Cool is trickling out on the heels of some recent legal maneuvering by Microsoft in its breach-of-contract lawsuit with Sun (Nasdaq:SUNW).

Earlier this month, Microsoft asked San Jose, Calif., Circuit Judge Ronald Whyte for permission to create a Java alternative that would not be bound by Sun compatibility tests.

Whyte hasn't ruled yet, although he termed the concept "very interesting." The issue likely won't be resolved for several months.

Late last month, Microsoft issued a patch that includes Sun's Java Native Interface to the Visual J++ development tool, as Whyte ordered in November.

If Cool is Microsoft's answer to Java, it creates even more questions for IT developers, such as Microsoft's status as a Java licensee and the extent to which it will attempt to make Cool cross-platform, as Java is.

Java Losing Momentum?

One developer believes, and hopes, that Cool will not be cross-platform.

"There are a lot of developers who say Java has lost momentum," said Mike Sax, president of Sax Software Inc., in Eugene, Ore. "If anything, making [Cool] not cross-platform seems like a smaller risk than going along with the Sun ruling and letting Sun call the shots."

If Microsoft makes Cool a product, the company will have a sizable challenge in gaining market acceptance, other developers said.

"Java is entrenched and is appreciated by a whole group of people out there who don't want to be tied to Microsoft for everything," said the Windows developer. "The world doesn't really need an alternative to Java from Microsoft."