PLAINTIFF'S EXHIBIT 2650 Comes v. Microsoft

From: John Ludwig
Sent: Monday, February 10, 1997 7:19 AM
To: Michael Toutonghi; Paul Gross; Brad Silverberg; Ben Slivka; Patrick Dussud; Bob Muglia; Erich Andersen (LCA)
Cc: Andrew Layman; Tracy Sharpe; Russ Arun; Nat Brown; Mike D. Smith; Anders Hejlsberg
Subject: RE: Friday, 2/7 Meeting with Javasoft

Categories: Java, Cabinet

given the flux in the sun side re native code interface, clearly they should have no business objection to removing the tests from the jct

-----Original Message-----
From: Michael Toutonghi
Sent: Sunday, February 09, 1997 3:43 PM
To: Paul Gross; Brad Silverberg; John Ludwig; Ben Slivka; Patrick Dussud; Bob Muglia; Erich Andersen (LCA)
Cc: Andrew Layman; Tracy Sharpe; Russ Arun; Nat Brown; Mike D. Smith; Anders Hejlsberg
Subject: Friday, 2/7 Meeting with Javasoft

TracySh, AndrewL and I met with Graham Hamilton, Jerome Douchez, Hans Muller, and Roger Riggs of JavaSoft to discuss serialization and object persistence specifications. I also met individually with David Bowen and Shen Liang about Sun's native code interfaces. Both meetings were quite productive. Since I was absent for part of the serialization meeting, if I omit anything significant, I would appreciate that AndrewL or TracySh add it.

NATIVE CODE MEETING

Key points

- Their interface is worse (less threatening) than I thought.
- They expressed **serious** interest in supporting COM.
- They are working on another low level native interface (LLNI) with advanced GC support (write barrier) that we may want to support if we participate in the design.

I did learn more about Sun's JNI native code interface than I had previously known. The most important thing I learned is that the interface is not very suitable for writing native components and less of a threat to native ActiveX development than I had begun to suspect. They have actually tried to make the VM linking model similar to COM, but have introduced more burden on the developer to use the actual API. We may be able to implement a COM based interface with some changes to theirs and ours. They seemed open to discussing a merging of APIs between JNI and COM. I suggested that we may want to consider something like this, but set expectations for possibly another version out.

They *seemed* genuinely interested in COM. David Bowen asked about licensing COM sources for including COM support with JDK releases. They we're mainly interested in the core pieces with a slimmer form of automation such as the core MAC COM support. If we could provide them with this and use COM as a native API, that would strengthen native ActiveX, but they may be blowing smoke. We should engage with them if it makes sense. Do we have a small, fast, COM core similar to the MAC versions at all?

They're working on another, low level native interface (LLNI). This could be part of the key to defusing this issue. This interface is more clearly targeting our raw native interface (RNI). They'll be supporting a write barrier for GC (as will we in the next version), and should have all of the basic functionality we provide in our raw native interface. Merging for compatibility at this level could be very beneficial to us. Developers targeting this level of development should be much fewer than COM, and are specifically writing Java supporting native code. Supporting this may mean depracating our RNI, but developers coding at that level should be able to deal with that. Shan Liang will be sending me a copy of their current spec which they said had not been seen yet by other licensees. They will probably release it to other licensees simultaneously.

They left with the following action items:

- Send a copy of the LLNI spec
- Discuss the possibility of supporting COM

I left with the following action items

- Look at the merging of their JNI interface with COM and discuss our interest in supporting JNI or the merged API
- Find out about what licensing they can participate in for COM

SERIALIZATION MEETING

Overview

- They were stalling numerous companies on addressing serialization until prompted by this meeting.
- We discussed requirements and agreed on most of our requirements.
- They have an approach in mind which is similar, but not as complete as the proposal we presented.
- They're willing to define something between our design and theirs as a standard way to persist through more of a property bag approach.
- **They're interested in using OLE structured storage.**

This meeting included discussing persistence requirements (easy ActiveX integration, version resiliency, implementation abstraction, property based persistence, text based persistence, structured storage based persistence). AndrewL brought a Microsoft proposal for persisting components to and from Java property bags. This proposal included a method of compatible interaction with the existing serialization model, and an improved, compatible persistence model with ActiveX integration. While Roger Riggs had been thinking along similar lines with a different implementation, they seemed to be very interested in the MS proposal. We all agreed that something which solved our mutual requirements should be decided on soon. They will reply with a merged proposal within 2 weeks. They also requested permission to discuss our proposal with Lotus and others, which we granted. Finally, we mutually agreed that we would inform each other of any major change in either of our persistence APIs while we continued working with them on the issue.

They're working on their own structured storage effort, and are interested in possibly using OLE's. Hans Muller had been asking previously about licensing the C source for structured storage. He brought it up himself and said that he wanted it because they are considering it as a standard format. This would be a major coup if we could somehow negotiate and license our way into using OLE as a common structured storage format for Java. While this is currently being treated as a small, peripheral issue, I believe it could have major ramifications.