From: Paul Gross
Sent: Monday, February 03, 1997 10:05 AM
To: Ben Slivka; Todd Knoblock; Charles Fitzgerald; Sara Williams (DRG); Daniel Weise; Peter Kukol; Anders Hejlsberg
Cc: Peter Plamondon; Robert Welland; John Ludwig
Subject: RE: potential issue for Java Language meeting

Ben,

As someone who has joined MS recently, I can tell you that when MS cries "Sun-proprietary" it isn't credible. Some other vendor might be able to get away with it, but not us. Second, having worked to push Sun on a single language change (successfully), I know how difficult it is to get them to move. They react best to highly scientific process on language changes. They want a written spec for multiple people to comment on and lots of data as to why language features are so important. They believe that because the language/VM have such widespread adoption that changes (even to ensure its future acceptance) have to be beaten down and only added annually.

I believe that our true goal, controlling the future of Java, will be totally transparent and mostly unacceptable to all Java OEMs. I believe that we currently have the most control and at some point will have total control. But taking control in this forum or this early stage will only alienate us. I think we are better off working issues with the likes of Borland, Symantec, Metrowerks, etc. and using them in addition to us to apply pressure. But the group effort organized by us smells bad.

Paul

—Original Message—
From: Ben Slivka
Sent: Sunday, February 02, 1997 12:04 AM
To: Paul Gross; Todd Knoblock; Charles Fitzgerald; Sara Williams (DRG); Daniel Weise; Peter Kukol; Anders Hejlsberg
Cc: Peter Plamondon; Robert Welland; John Ludwig
Subject: RE: potential issue for Java Language meeting
Importance: High

My goal for this meeting (said goal having evolved over the past few weeks) is to start a dialog about the evolution of the Java language. Sun has resisted putting Java into a standards body, and I think this meeting will help push them toward opening up at least the language more quickly than they had planned. I'm sure someone could paint our involvement in this coalition in a sinister color, but our rejoinder is very simple. We think Java is an important technology, it is Sun-proprietary right now, Sun has described their strong desire for a "free language, as free as English, French, and German", and we think Sun needs to follow through on this pledge.

I don't see any downside to holding this meeting on Friday, I'm confident our spin master (charlesf) will be able to stay on top of the JavaSoft folks.

Paul, can you provide specific examples of negative outcomes for Microsoft from holding this event, given that we characterize our interests as above?
Thanks,
—bens

—Original Message—
From: Paul Gross
Sent: Friday, January 31, 1997 7:10 PM
To: Todd Knoblock; Charles Fitzgerald; Sara Williams (DRG); Ben Slivka; Daniel Weise; Peter Kukol; Anders Hejlsberg
Cc: Peter Plamondon; Robert Welland; John Ludwig
Subject: RE: potential issue for Java Language meeting

MSS 0056495
CONFIDENTIAL

Comments below...

—Original Message—
From: Todd Knoblock

Exhibit No. 8
Wit: Gross
Dorothy Stiles, CRR
6/25/95

GOVERNMENT EXHIBIT
1326
PENGAD-Bayonne, N. J.

Sent: Friday, January 31, 1997 12:56 PM
To: Charles Fitzgerald; Sara Williams (ORG); Ben Slivka; Daniel Weise; Peter Kukol; Anders Hejlsberg
Cc: Peter Plamondon; Robert Welland; John Ludwig; Paul Gross
Subject: RE: potential issue for Java Language meeting

Java is all things to some people and some things to all people. One practical question is whether to consider the standard libraries (java.lang.*, java.awt.*, etc.) as part of the "language" or not. Changes (including additions) have all the same repercussions as changes in the core language per se. Most languages have a "standard" library that is considered part of the language: c and stdlib, C++ and STL, Common Lisp & built-in library. However, for the purposes of this discussion, I will restrict my attention to the core language, i.e., linguistic constructs sans libraries.

I would be reluctant to suggest major changes to the Java language for several reasons: ---

1. Java is a calculated compromise between expressive power and simplicity. Any change that increases the expressivity must be weighed against the increase in complexity. This is, at best, a judgment call and can not easily be argued on technological grounds alone.
2. As designed by a small core of designers, it has thus far avoided the design-by-committee problems that plague too many languages. While certainly frayed around the edges, I think we can all agree that it is an okay design, and could be made worse.
3. Politically, I don't think it will serve us well with the community if we are viewed as "cooping" or "spoiling" the Java simplicity of design.

[Paul Gross] I would state this more strongly. I believe that this meeting has the potential to totally backfire on MS. It is very transparent to JavaSoft and they heard about it from Mansour before we called James (mistake #1). Mansour characterized similarly to Sara's definition of the goal. I believe this transparency could end up getting picked up by the press and would hurt us with the 3rd party community and with our potential customers. I think we should seriously re-consider whether this event will do more harm then good and whether we need to take this approach to win in either case. I would argue that our execution is our single strongest asset (beyond Windows market share) and we should just build better VMs, class libs and tools.

Nether the less, where the design incurs significant inefficiency which can be repaired without introducing undo complexity, there is room to tinker. The following lists my ideas for things that should be changed, things that should be considered for change but not changed immediately, and things that should not be changed.

- **Consider changing soon**
  1. Finalize method semantics. The current semantics are too vague, and a great possible source of differences in run-time semantics between VM implementations.
  2. Error exception semantics. The current order-of-exception requirements are too constraining and virtually rule-out significant compiler reordering of computations (e.g., hoisting checks out of loops).
- **Consider adding**
  1. Parametric polymorphism. The otherwise sound static typing cries out for a way to parameterize classes.
  2. Algebraic data types. Stronger support for datatype constructors could be considered
  3. Lambda. Higher order functions would be useful, and should be weighed against complexity,
  4. Assertions. Presently there is no way to easily write assertion-like code in Java. Consider a standard, C style macro assertMessage(a[i]==null, "Expecting null, found"+a[i].toString()). This will fail in java if AssertMessage is written as a function because the second argument would be evaluated even when a[i] was null (i.e., because of the call-by-value semantics of java). If assertions were primitive, this problem is moot.
  5. Remove inner classes.
  6. Configuration management. It is difficult to control multiple configurations for, e.g., Debug versus Retail builds under Java. Like assert, there is no convenient way to arrange for conditional compilation under java. The obvious use of static final boolean fields does not suffice because the compiler will insist that all the code be compiled before half of it is discarded.
  7. Localization. Is the current JDK 1.1 spec. sufficient for MS.
- **Don't add**
  1. Macros. To my mind, if the idomatic uses of macros are covered using another mechanism (e.g., conditional compile, assertions, localization), then macros *should* be left out. It makes the code greatly simpler to read and understand. It also lets us build better editors and debuggers.

MSS 0056496
CONFIDENTIAL

This is a good trade-off in the expressivity versus complexity front.

2. Templates. Templates are a poor implementation of parametric polymorphism. There are better ways.

Todd

---Original Message---
From: Charles Fitzgerald
Sent: Friday, January 31, 1997 12:29 PM
To: Sara Williams (DRG); Ben Slivka; Daniel Weise; Peter Kukol; Todd Knoblock; Anders Hejlsberg
Cc: Peter Plamondon; Robert Welland; John Ludwig; Paul Gross
Subject: RE: potential issue for Java Language meeting

We do need to flesh out our agenda, even if there is no official agenda. from Ben's original list:

- Inner classes - just say no?
  - Add Closures instead?
- Conditional compilation
- Enumerations
- Add properties
  - Would permit associative arrays, other Python features(?)
- Debug class
  - Asserts
  - Debug outs
    - Final members
    - Compiler can remove the call to empty classes for retail build
- Moratorium on language changes for 1-2 years (Symantec suggested this)
- Floating point support

What other topics should be on the list? We need to get our agenda nailed pretty quickly so Peter has time to work selected issues with other attendees in advance.

---Original Message---
From: Sara Williams (DRG)
Sent: Friday, January 31, 1997 12:19 PM
To: Ben Slivka; Charles Fitzgerald; Daniel Weise; Peter Kukol; Todd Knoblock; Anders Hejlsberg
Cc: Peter Plamondon; Robert Welland; John Ludwig; Paul Gross
Subject: RE: potential issue for Java Language meeting

i agree, although we don't have a set agenda. charles - do you want to invite him since you know him, or do you want me to?

-Sara

---Original Message---
From: Ben Slivka
Sent: Friday, January 31, 1997 10:17 AM
To: Charles Fitzgerald; Daniel Weise; Peter Kukol; Todd Knoblock; Anders Hejlsberg
Cc: Peter Plamondon; Sara Williams (DRG); Robert Welland; John Ludwig; Paul Gross
Subject: RE: potential issue for Java Language meeting

I think it would be good to invite him, limit the agenda to ½ hour on this topic.
--bens

MSS 0056497
CONFIDENTIAL