First Ever Smalltalk-80 Implementors' Conference
September 24-25, 1981

Minutes

(These minutes are intended to jar your memories, rather than to be a pretty record of what went on, hence their form.)

Introduced everyone
Went over goals/agenda

Technical Area -- Termination of Review Process

Discussed "September 30" as date on which last image would be frozen
Tape would come a couple of weeks later to allow for documentation
Strong desire to make few/no changes to VMachine from June 30
Strong desire to make this be clean, even at the expense of slipping schedule
Book (vol. 1) will go to publisher on Dec 31; (by definition); review will be done at this time
Should image have changed (desire not to); reviewers will receive new image, which will be the license image (if any)

Image release process bugs/changes

Interchange Format:
Discussed format for images with words other than 16 bits; decided since it is not specified what is needed, we will stick with the one we have.
Discussed whether we would release a small image, with various things clamped out; decided we would not, but might provide guidelines somewhere in the system for paring it down.
Discussed snapshot format as distinct from interchange format; decided this was ok, but required primitives
primitive a) snapshot to snapshot image; required
primitive b) snapshot to interchange format; optional
plus defined starting Smalltalk-80 as resuming snapshot format
To allow an interpreter to discover whether a file was interchange format or not, interchange format was amended to insist that seventh word be 0 for interchange format, non-0 otherwise

System bugs/changes

String copy:
Discussed adding primitives for String copy, and for String compare, because of measured effects on Dolphin.
Discussed using BitBlt instead.
Decided probably this was only one of several places that, depending on the implementation, would need to be sped up; therefore, it would be up to each implementation.

Process primitives:
Discussed whether to make them to be stack rather than queue; did not resolve

CursorLink:
Whether true/false at startup, also discussed the state of the system in general at snapshot resumption/startup. Decided it should be further specified.

Display Size:
Discussed minimal/interchange (startup) display size; did not resolve, but probably should be 640x480 or less and allowable size should/could be determined by a primitive

Larger OT and Object Spaces:

Discussed their ramifications, besides interchange format, also large/small integers, and how this relates to system specification; no resolution

Mouse Bounding:
Whether the cursor could leave the display, and if so by how much; discussed importance to graphics world; did not resolve

BeCursor:
It does not affect the position. What is returned should be specified.

BeDisplay:
Discussed beDisplay especially as it affects remote displays. Noted that having a remote display is a form of caching, and should have the appropriate treatment; in particular, BitMaps are currently accessible only through BitBlt and at:/at:put:, so those primitives must know about the cache.

Initialization of Methods:
Methods should be initialized through the primitive, but the system did not reflect this on June 30. Use of 0 for initial bits is bad for literal frame, 2 (nil) would not hurt.

Perform with arguments:
Could possibly go off the end of the stack in its current specification. Could activate without pushing on current stack first, but this has problems. Compiler could force a large context when perform:withArguments: was sent (although this would only reduce not eliminate probability of failure)

SomeInstance/nextInstance Primitives:
Fail code should be specified.

Reference Counting:
Discussed ramifications of reference counting, such as storing nil when popping the stack, what happens with perform:withArgumentss:, and the process primitives. Decided code in book would be too cluttered to include it all, but people should be careful.

Instantiation of >1 object at a time, as in the messageNotUnderstood code, could cause problems, because it involves storing references to one object in another that may not yet officially "exist" (because it has no references to it yet); again one must be careful

Discussed whether implementors could count on not reference counting various things, such as small integers, and the canned oops. Decided that it was ok to ignore small intergers, probably fine to ignore nil, true, and false, but not really good practice to ignore the others, but let the overflow bit work, and optimize that if necessary.

The @ Primitive:
Specification should be changed to fail for non-SmallIntegers rather than non-Numbers.

**Quo:, rem:, div:, mod:, /, //, \\, **
Discussed that the world has at least two consistent views of integral division and remainder that differ in the way they deal with negative remainders, and that often people implement division from one view and remainder from the other. To satisfy everyone, all possibilities are included in the Smalltalk-80 system.

0 vs. 0.0 in Float code:
Some code would be more optimal if it used 0.0 rather than 0 because of coercion.

Source Code Management:
Discussed current strategy as space-efficient, but dirty.

Discussed that the real solution is to include source code as Smalltalk objects in a virtual memory system, rather than trying to use an external thing (files).

Discussed saving temp names and comments only, rather than all of the code and having a smarter decompiler

Semaphores and Errors:

Discussed out of memory/oops semaphores; decided not to require it.

Discussed the general use of the semaphore mechanism as standard way to signal errors, implementation-specific

Discussed one of these semaphores perhaps being for invalid bytecode, which should never happen, of course

Discussed out of memory condition, whether one could recover in time, and what would be involved. (e.g. soft and hard storage limits).

Discussed that it would be nice to have a convenient facility for adding handlers for various other machine exceptions

System Optimization:

Discussed that new system will have different profiles, should be optimized here as much as reasonable, then up to implementors

Primitive Failure Reporting:

Discussed giving more information to Smalltalk code on why primitive failed; decided this was in the realm of multiple-object returns from methods, and therefore a language research question.

Known/future bugs:

Discussed compatibility vs. extensions, centralized control vs. decentralized, future vs. present

Technical Area--Future Collaborations

Virtual Memory as First Instance

Outlined LOOM design, discussed its transfer (if any) as example of future collaboration

Other stuff

Discussed what benchmarks to use to compare (if desired) implementations or to measure progress of one implementation; collection to be included in a Smalltalk-80 class definition; some attatched. Benchmarks seem to come in "macro" and "micro" styles: e.g., speed of executing 3+4 as micro, and speed to browse as macro. Both would be good

Applications

Areas for major extension, such as multiple superclasses

Non-Technical Areas

Motivations for implementing

Prototyping, analysis of system

Sustaining interest (assumed)

Collection of papers based on experiences

Software: Practice and Experience suggested, desired by some as "real journal, so can justify writing to management"

Our own paperback book, makes it easier to write

Adele volunteered to begin organizing it

First drafts due around February, to give enough time to reflect, but still fresh; final drafts to publisher in Spring

Topics include: "The Xerox Story of Smalltalk-80", "what I did", "what I would have liked to have known that I know now", "what I should have done", and many more

Discussed including (if ok with authors) PASCAL versions of interpreters; transliteration of book vs. transformed for optimization, both would be of interest

Topic could be "how to write a Smalltalk-80 program"/"what is a program in Smalltalk-80"

Future Newsletter

Allan Shiffman from Fairchild wants to make Smalltalk magazine or newsletter
Contributions could include technical issues with solutions sought
Could be forum for challenges in applications/design/language ideas/...

Future Implementors'/Users' Groups

Too early for Users'
Maybe in Spring (as part of the process of finishing the book/collection of articles)
Tek people volunteered Portland

Smalltalk-80 Licenses/Licensees

Adele outlined proposed Xerox licenses for Smalltalk-80; should be available 1st quarter 1982, when book goes to publisher; licensees get pre-publications book copies
Licensees should be involved in future implementors' meetings asap

Other Topics

Writing an article for managers; for *WSJ* or *Forbes*; discussing issues of how Smalltalk aids (does not aid) "productivity" in design/prototype/programming/...
Discussed blocks; recursion and its problems, sharing temps, syntax of args
Mouse buttons and their use; will be three (three states) in interface
Discussed typing as method of browsing