LARRY Sutter

Geo.

Tektronix®
COMMITTED TO EXCELLENCE

INTER-OFFICE
COMMUNICATION

(F)

Smalltalk

Retsc

Thanks

TO: George Rhine

DATE: June 3, 1981

FROM: Paul L. McCullough

SUBJECT: Smalltalk - 80 and NCC

On May 6 I traveled to Chicago to participate as a panelist on an NCC panel, "The Public Release of Smalltalk - 80". My trip report is divided into four sections.

I. The NCC Exhibits

The amount of time that I had available for viewing exhibits was limited. I am sure that I missed alot. The booths of major interest to me were:

Tektronix

The Tek booth was impressive. I had not previously seen the 4112, 4114, or eight-pen plotter in operation. I talked to several Tek field people. They are quite concerned about Jon Reed's statements about color graphics, and they want to know when the follow-ons for the 4050 series will be available.

Apple

The Apple booth consisted of demonstrations of Apple II and Apple III. No one from Apple is saying anything about Apple IV, but they seem very positive that the manufacturing and software problems related to Apple III are solved.

Three Rivers

A very small booth, demonstrating their personal computer, the PERQ. Similar in concept to the Xerox Alto, the PERQ is many years behind Xerox in the areas of human interaction and visually pleasing display of information. Adele Goldberg of Xerox did mention that Three Rivers would like to obtain a Smalltalk - 80 license.

Xerox

By far, the most popular booth at the show. The most commonly heard phrases on the display floor were "Xerox" and "Star". Star is the newest Xerox product, and the first product to show many aspects of their investment at Xerox' Palo Alto Research Center (PARC). Over the past decade, Xerox PARC has developed a number of proprietary processors. Alto, their first machine, eventually had 1,200 machines installed. Several years ago, their researchers realized that the Alto had reached its limits, and initiated the Dorado (or D1) project to develop an experimental, very fast, personal computer. Only a few Dorados were expected to be built, but plans were made to build slower, more compact machines, the Dolphin (or DO). The Dolphins are replacing the Alto within Xerox, and appear in at least one product, the 5700, which acts as a laser printer and electronic mail server. Somewhat under the table, artificial intelligence researchers have been purchasing Dolphins to run Interlisp. Yet another processor, the son of Dolphin, is the processor used in the Star.

The Star salespeople stress that the Star is a workstation for professionals - not a word processing system. The user interface is modeled closely on the user interface of Smalltalk. Via an Ethernet, multiple Stars may be inter-connected, as well as printer servers, file servers, and communication servers.

I have good copies of the Seybold report on Star available.

II. The Smalltalk - 80 Panel

The panel consisted of:

Daniel Ingalls, Xerox PARC, Learning Research Group (LRG)
Larry Tesler, Apple (formerly of Xerox PARC LRG)
Dave Robson, Xerox PARC, LRG
Ted Kaehler, Xerox PARC, LRG
Ted Laliotis, Hewlett-Packard, HP-Labs
Trygve Reenskaug, Central Institute for Industrial Research, Oslow, Norway
Stoney Ballard, Digital Equipment Corporation, Corporate Research
myself, TEKTRONX

There were approximately 500-600 attendees. Each panelist gave a 10-15 minute talk about different aspects of Smalltalk - 80. I will give a brief synopsis of each panelists presentation.

Dan Ingalls

Dan presented a brief introduction to Smalltalk - 80 (object-oriented system, etc.). Major points:

- Allows personal mastery of the system
- Allows user to exert maximum leverage

Larry Tesler

Larry's presentation consisted of slides and a talk illustrating aspects of the Smalltalk - 80 user interface and programming environment. Major points:

- Reactive system - not just interactive. All objects in the system are ready to function at all times.
- A strong synergy between the programming language and its environment.
- Highly integrated system
- Excellent user interface

Dave Robson

Dave described the distribution process (the book and the virtual image) and briefly discussed the interpreter. Major points:

* Virtual Image is a good way to distribute the system, and achieve a standard base.
* Objects can be viewed as "packets of functionality"

Ted Kaehler

Ted discussed the virtual memory system in Smalltalk - 76. I believe that his article in the August issue of *Byte* will describe the Smalltalk - 80 virtual memory. Major points:

* V.M. is implemented entirely in Smalltalk
* About 90K bytes of real memory provides a system with quite acceptable performance

Ted Laliotis

Ted is the manager of implementation for Smalltalk - 80 at H-P Labs in Palo Alto. There is also a one-person project at Fort Collins. He said that H-P Labs is using an HP-44. Thinking back, I am sorry I didn't ask exactly what that is. H-P looks upon their Smalltalk - 80 effort as an evaluation of the system as a personal computing environment for *non*-accomplished programmers. He also mentioned that they view Smalltalk - 80 as a product useable by non-hackers. There seems to be a dichotomy here. Major points:

* Disclaimer: HP is not promising Smalltalk products.
* BASIC is grossly inadequate as a language.
* They see Smalltalk - 80 as having good potential in an office environment with a high-resolution color display.
* Their view of a personal computing language: object-oriented and highly interactive.

Trygve Reenskaug

Trygve is the original implementor of Smalltalk outside of Xerox, having implemented Smalltalk - 78, on a Norwegian version of an 8086, as Xerox' first test site. He is *very* positive about Smalltalk. Major points:

* If you're fed up with your present system, if you're tired of fighting with your system to make it do what you want, then try Smalltalk - 80 -- you'll be pleased.
* He gave an excellent analogy, with present systems, it is as if you are looking thru a keyhole. You may see certain art treasures, but from a very limited perspective. With Smalltalk - 80, you unlock and open the door, allowing you to enter a new world.

Stoney Ballard

Stoney is implementing Smalltalk - 80 on a PDP 11/23, and estimates that it has taken about four man-months. There is another effort at DEC to implement on a VAX, but Stoney has little hope for a successful project (and declines to say why). He is also very positive when speaking on Smalltalk - 80. Major points:

- Disclaimer: DEC is not promising Smalltalk products
- Remarkably easy to implement
- Interpreter easy to design and code
- Object-memory manager more difficult to design

A synopsis of my talk is attached.

The question-answer period was brief. The major points that were brought out:

- The book will be ready to publish and the virtual image ready to distribute at the end of summer.
- Licensing fees for Smalltalk - 80 are not yet established.

After the panel session, Christopher Morgan, Editor-in-Chief, of the Byte magazine approached me about quoting some of my comments in the August issue of Byte magazine (the August issue is devoted to Smalltalk - 80). Provided that he gives a disclaimer that Tektronix is not promising to produce Smalltalk products, I see no problems.

The trip was certainly worthwhile. It allowed me to meet the other implementors and some of the Xerox designers and implementors, while performing some market research and tossing Tek's name into the Smalltalk - 80 arena.

Because of the need to finish GCS Pascal, we are definitely behind the other implementors. DEC and H-P openly state that they have the system running, and I know via a friend at Apple that they are quite far along. These companies have been able to dedicate personnel to their Smalltalk --80 projects.

III. Some Recommendations

Given the cost of the 4112 and the proposed cost of the 4061, I recommend that Tektronix investigate the possibility of entering into an OEM agreement with Xerox for the Star hardware and supporting software (Mesa programming language, Pilot operating system, etc.). Given these hardware and software building blocks, Tektronix may be able to build new products at a lower cost.

I would also encourage Tektronix to acquire a Star system or two, both to use as a workstation and as a tool for studying user interfaces.

IV. A Trip to Xerox PARC

While speaking to Xerox personnel, it became evident that they would very much like us to spend a day at Xerox PARC. The day's activities would include:

* Smalltalk - 80 demo
* Discussions with the designers of the system
* Exchange of ideas.

I would certainly encourage such a trip.

A list of attendees might include: Allen, Larry, Jack Grimes, George Rhine, and myself.

PM:jc

cc: Tom Cook
Jack Grimes
Tom Hamilton
Dave Heinen
Larry Katz
Robert Reed
Rick Samco
Allen Wirfs-Brock

George Rhine
Paul McCullough
Smalltalk - 80 and NCC
June 3, 1981

SUMMARY OF MY TALK

I. Why Tek is Interested

A. Provides an opportunity for Tek to take advantage of a decade of research, experience, and hard work by the folks at Xerox.

B. Allows us to evaluate an object-oriented language and system.

C. Tek is a graphics company, whether the graphics happen to be waveforms, financial information, or energy research data, Tek is interested in novel ways to collect, manipulate, store, and display information, and Smalltalk - 80 appears to be an excellent source of information.

D. We will be investigating the applicability of Smalltalk - 80 across several divisions of the company.

II. From a Software Engineering Viewpoint

A. Software engineering is usually viewed as consisting of:

a goal
a design methodology
a structured language with which to implement the design, and
constraints imposed by the goal, the design, and the language.

1) Many current design methodologies insist that the design be complete before coding begins. Unfortunately, should the goal be changed slightly, or should a constraint change, changing the design, or worse, changing the code can be very difficult.

2) Smalltalk - 80 encourages design via its modularity, as well as evolving design - that is, the software engineer is encouraged through the interactivity with the system to experiment with several designs, and then choose which best meets the goal.

3) Structured languages, such as Pascal, provide the software engineer with the ability to declare new types and encourages strong type-checking, but lacks the ability to declare new operators or to define new operations for existing operators.

4) Smalltalk - 80, encourages the user to declare new classes of objects, and provides the ability to declare new operators and redefine the meaning of operators in controlled ways.

B. A software engineer may view a large project as hundreds of sessions lasting several months at a terminal. With ST-80, the design and implementation is one long session, lasting months perhaps, with all objects in the system current and active and ready to perform their function.

C. A good keyword for ST-80 is "uniform".

1) Uniform reference - all the details of how a data structure is implemented are enclosed within a class description, and all operations on an object must be made via its message interface. All the code to manipulate an object is in one central place, the class description.

2) Uniform representation - the LISP language achieves incredible mileage out of one simple fact: everything in a LISP system (data, functions, compiled functions) has one representation: the LISP cell. Everything in LISP (including the compiler, editor, debugging tools, and what have you), everything is built upon this one uniform representation.

ST-80 also has one uniform representation - everything is an object. The entire system is built upon this one simple idea. Everything is an object - including the editor, the compiler, even the display.

3) Uniform invocation of operators- to invoke an operation, a message is sent to an object.

4) A simple, uniform user interface.

D. As we have mentioned, ST-80 provides very good encapsulation of data structures. Some other languages that support encapsulation are CLU, MODULA-2, and ADA. Each of these has a relatively short record of use, Smalltalk - 80 and its ancestors have nearly a decade of use.

E. ST-80 provides many building blocks to the software engineer.

As an example, consider what must be done in a conventional system by a software engineer who is constructing a program that requires a symbol table. The engineer must design and write routines that insert and delete entries from the symbol table, and routines that can report or change specific fields of a symbol table entry. These routines might use a hashing function which the engineer would also have to write.

With ST-80, the software engineer would simply create a new object of class Dictionary (a part of the standard ST-80 system). Objects of class Dictionary already know how to insert, delete, and hash entries. Much of the low-level support that the software engineer would have to design and code in a conventional system is provided by the ST-80 system.

F. Smalltalk - 80 is a good simulation tool. Painting, animation, music, and other simulations have been developed with Smalltalk - 80 predecessors. Even if a final product will not be implemented in Smalltalk - 80 (for size constraint, for example), much design and trial efforts can be done in Smalltalk - 80, thus speeding the development of the final product.

G. Perhaps most importantly, the software engineer need never leave the Smalltalk - 80 world: the system provides tools for designing, compiling, editing, and so on.

III. Conclusion

Fred Brooks tells us that when designing a system, we should "plan to throw one away". Over several years Xerox researchers have designed and built several ST systems and they have gained a lot of experience; experience in language design, graphics, and user interfaces, and this experience is now being made public.

I personally believe that ST - 80 will:

* provide insight into new ways to use computers,

* change the ways we interact with computers,

* change the ways we think about computers,

* and that it will have a broad impact on the computer industry.