Smalltalk at

Tektronix

STIC13
June 10, 2013

http://www.wirfs-brock.com/allen/things/smalltalk-things/tektronix-smalltalk-document-archive

Tektronix Invented the Precision Oscilloscope

Versatility...Plus

The Tektronix Type 511 is a portable wide band oscilloscope providing facilities formerly available only in very expensive, cumbersome instruments.

Tektronix Type 511 Oscilloscope

SWEEP CHARACTERISTICS

Continuously variable .1 second to 1 microsecond (10 cm. deflection).
Direct reading sweep speed dial.
Choice of triggered, recurrent or single sweeps at all speeds.
Triggers on sine waves to 10 mc. or pulses over .05 microsecond.
Any 20% of sweep may be expanded 5 times.
DC coupled PP amplifier for external sweep input.

VERTICAL DEFLECTION SYSTEM

Amplifier Bandwidth 10 mc., 1 stage; 8 mc., 2 stages.
Rise Time .04 microsec., 1 stage; .05 microsec., 2 stages.
Maximum Sensitivity .27 V/cm. (Peak to Peak).
Input Impedance Direct 1 meg., 40 mmf.; Probe 10 meg., 11 mmf.

MISCELLANEOUS

Calibrating voltage 0-1, 0-10, 0-100 volts, 60 cycles.
CRT 5CP1A, 5CP7A or 5CP11A operating at 3 kv.
Direct connection to all plates from side panel.
Total weight 65 pounds, self contained.

Price $795.00 f.o.b. Portland

Your inquiry will bring more detailed information and name of the nearest Field Engineering Representative.

Phone, EAst 4885
Cables, TEKTRONIX

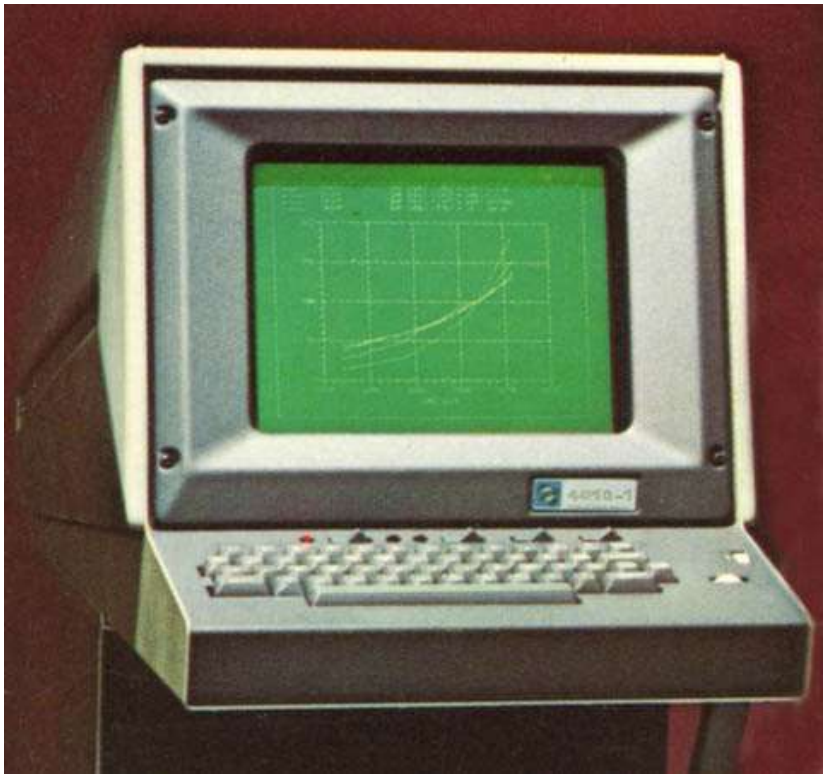
TEKTRONIX INC.

712 S. E. Hawthorne Blvd.
Portland 14, Oregon

ELECTRONICS — *September, 1948*

And by 1970 was one of the worlds largest and most important electronics companies.

In the Early 1970's Storage Tube Scope Technology Begat Graphics Terminals



Tek Graphic Terminals dominated the "low cost" computer graphics market for the entire 1970's decade.

Graphic Terminals Begat Desktop Computers

Introduced in October 1975, the Tek 4051 was arguably the first desktop graphic “personal computer”

Compare to IBM 5100 introduced in September 1975: http://en.wikipedia.org/wiki/IBM_5100

1980

- Tek's Graphic Computer System (GCS) was the business unit responsible for the 4051 and its successors products.
- Allen Wirfs-Brock and Paul McCullough were working for GCS building a systems programming oriented Pascal compiler for the Motorola 68000.

Tek 4041 BASIC Language 68000-based Computer Disguised as Electronics Instrument

GCS Pascal was used as systems programming language for these products.

Tek 4909 "networked" File Server

Rebecca Wirfs-Brock was lead SW Engineer

http://www.wirfs-brock.com/allen/files/tek/gcsPascal.pdf

Xerox Corporation
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
415 494-4000

July 25, 1980

XEROX

Mr. Jack Grimes
Tektronix Inc.
P. O. Box 500
Beaverton, Oregon 97077

Dear Mr. Grimes:

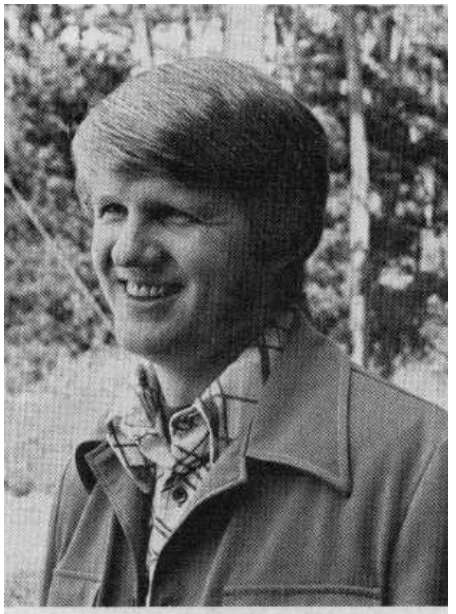
The Learning Research Group at PARC has been involved in the design and implementation of a software system for personal computing widely known as Smalltalk. After nearly a decade of research, we are preparing to publish a book documenting Smalltalk which will describe the language, its implementation, its text and graphic features, its document editors, and its program development and debugging tools. The Smalltalk-80 programming system is the latest version of this integrated environment for specifying, implementing and debugging complex software applications. User level access to the Smalltalk system is a highly graphical approach to viewing structured information that utilizes display screen menus and user selection of messages to obtain or change data. Smalltalk itself is a programming language based on an object approach to data representation and manipulation, and a message approach to processing.

Our main goal in publishing at this time is to disseminate widely a basic standard for Smalltalk as a language and system. We hope to foster compatibility between the anticipated personal computer versions of the language by providing definitive reference documentation. A further goal is to have a number of implementations of the system already completed at the time the book is available. By cooperating in these implementation projects, we hope to receive early feedback about needed revisions of the Smalltalk book draft and to promote even greater standardization of Smalltalk.

For purposes of review and basic implementation, we are inviting a limited number of manufacturers to participate in a "pre-publication review" of the Smalltalk book. To these manufacturers, we are proposing a two-phase review process:

Phase I - Evaluation of Written Material
Phase II - Trial Implementation

http://www.wirfs-brock.com/allen/files/tek/1980-7-25-invite-letter.pdf



Dr. Jack D. Grimes of Tektronix has been appointed technical editor of *Computer*. He succeeds Dr. Jack E. Shemer of Xerox in the post.

http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01647409

Phase 1 – Review the Book

Tektronix
COMMITTED TO EXCELLENCE

INTER-OFFICE COMMUNICATION

TO: List

DATE: 10-14-80

FROM: Paul L. McCullough

SUBJECT: Smalltalk Review

The first meeting of the Smalltalk review group will be Thursday, October 16, in building 63, conference room W-52, from 9:30 to 11:00.

The agenda is:

* Overall impressions concerning the portion of the book that we have received.
* Specific comments on the prologue, each chapter, and appendix.
* Summation of comments and impressions.
* Discussion of whether to return these chapters or retain them a while longer.

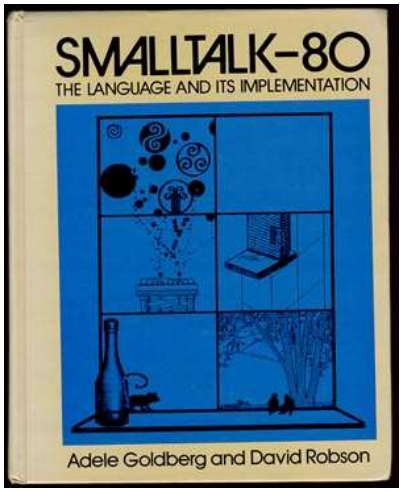
PM:jkb

List: Jack Grimes
Dave Heinen
Larry Katz
Bob Reed
Rick Samco
Don Williams
Allen Wirfs-Brock

SMALLTALK–80 PROJECT: PHASE I

* BOOK REVIEW FOR XEROX
* 12 CHAPTERS:
  + 10 RECEIVED
  + 9 REVIEWED
* 5 PEOPLE:
  + L. KATZ — R. SAMCO – TEK LABS
  + P. McCULLOUGH — GCS/DAD — R. REED – T&D
  + A. WIRFS–BROCK
* BEGAN SEPT. 1980

February 1981 Status

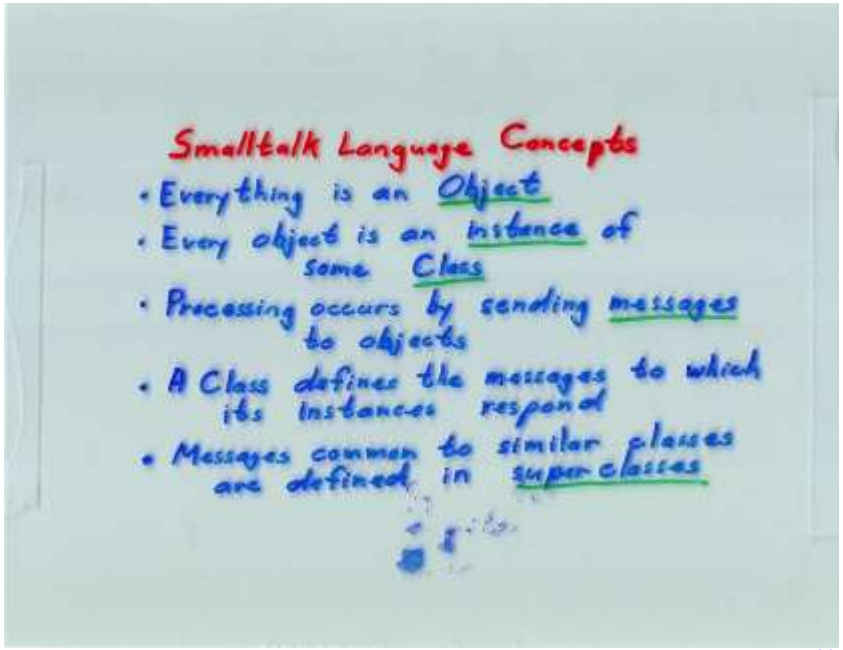


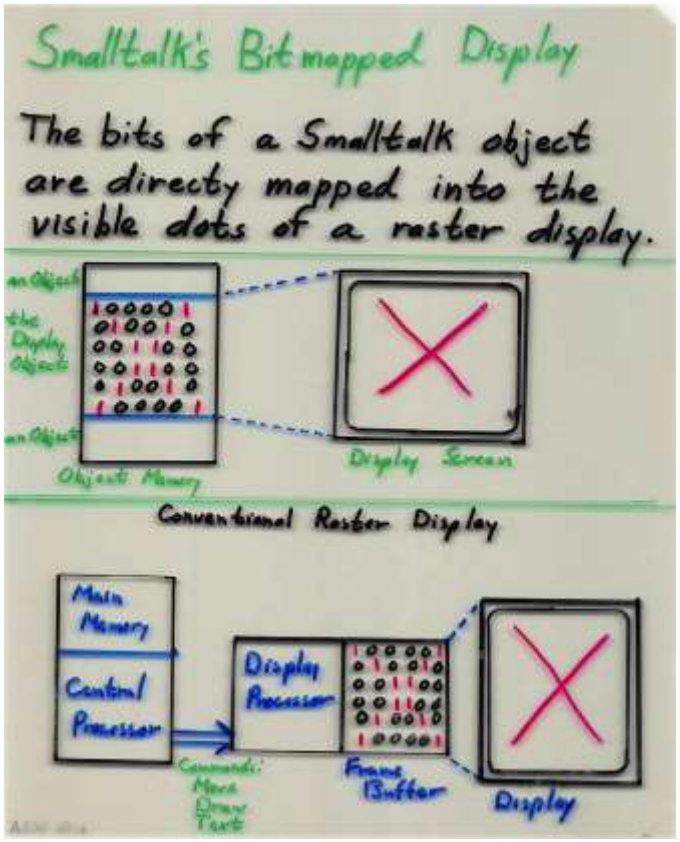
http://www.wirfs-brock.com/allen/files/tek/1980-10-14-1st%20book-review-meeting.pdf
http://www.wirfs-brock.com/allen/files/tek/1981-2-Larry-Katz-intro-and-status.pdf

Started Internal Smalltalk Evangelism

WHAT IS SMALLTALK–80?

AN INTEGRATED ENVIRONMENT FOR
SPECIFYING, IMPLEMENTING, AND DEBUGGING
COMPLEX SOFTWARE APPLICATIONS





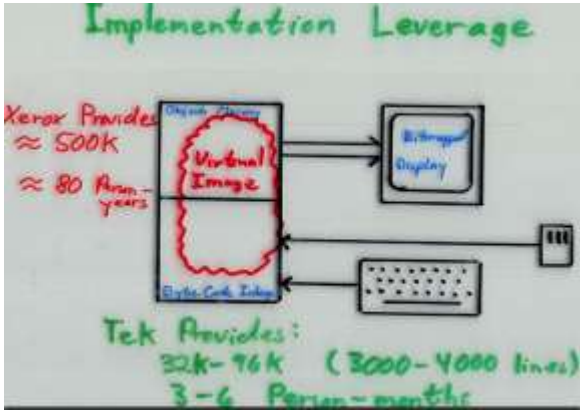
http://www.wirfs-brock.com/allen/files/tek/1981-11-pitch-deck.pdf
http://www.wirfs-brock.com/allen/files/tek/1981-2-Larry-Katz-intro-and-status.pdf

Phase 2 – Implement a Virtual Machine

SMALLTALK–80 PROJECT: PHASE II

- EXPERIMENTAL IMPLEMENTATION
- OBJECTIVES:
  - + PRODUCE A WORKING EXPERIMENTAL IMPLEMENTATION OF SMALLTALK–80
  - + VERIFY ACCURACY AND COMPLETENESS OF DOCUMENTATION
  - + PROVIDE FEEDBACK TO XEROX
- BEGINS NOW (FEB./MAR. 1981)

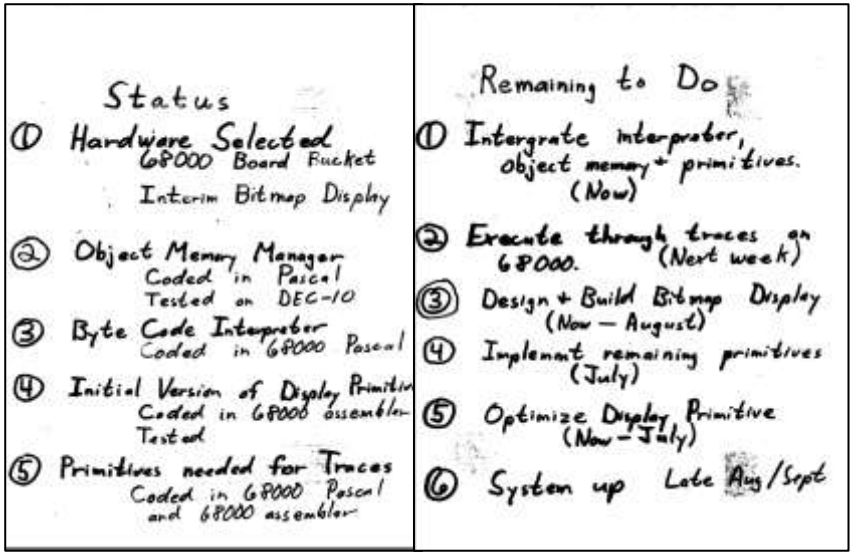
February 1981 Status



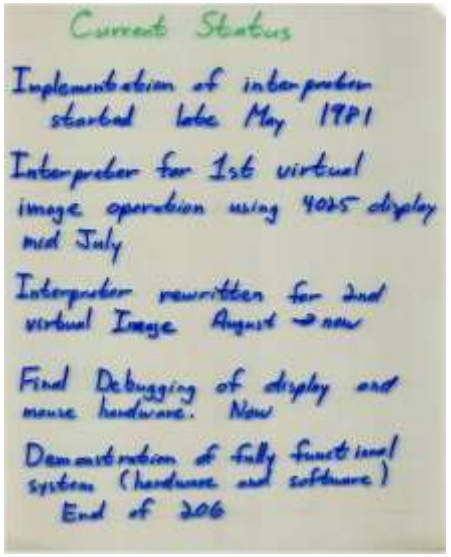
People who contributed: Paul McCullough, Allen Wirfs-Brock, Jason Penny, Allen Otis (hardware engineering), Larry Katz, Alan Purdy

http://www.wirfs-brock.com/allen/files/tek/1981-11-pitch-deck.pdf
http://www.wirfs-brock.com/allen/files/tek/1981-2-Larry-Katz-intro-and-status.pdf

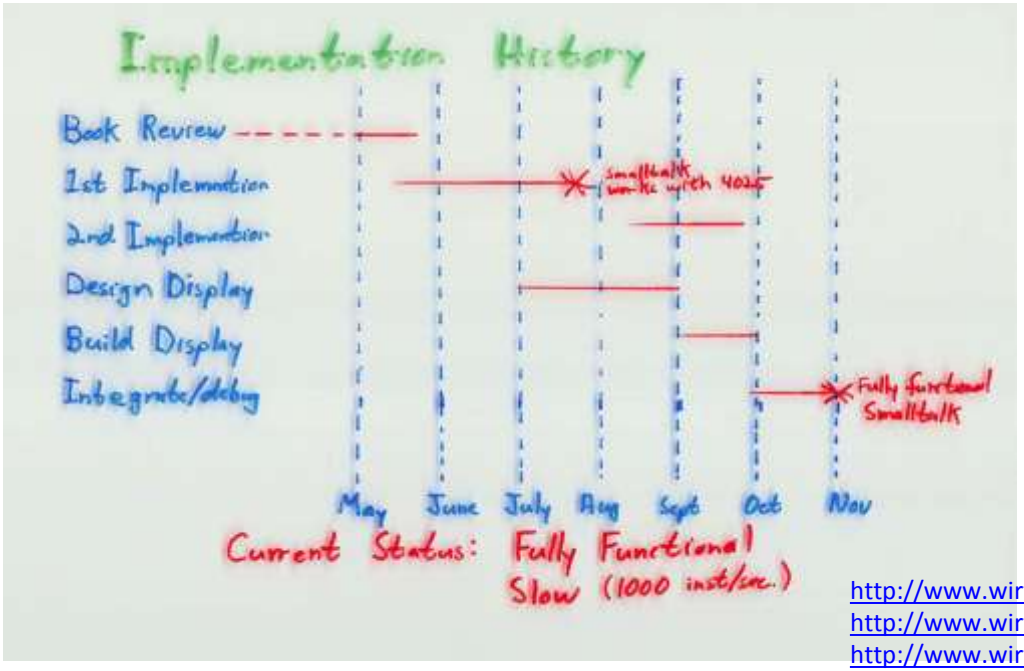
Phase 2 – Implement a Virtual Machine



Late May/Early June 1981 Status



September(?) 1981 Status

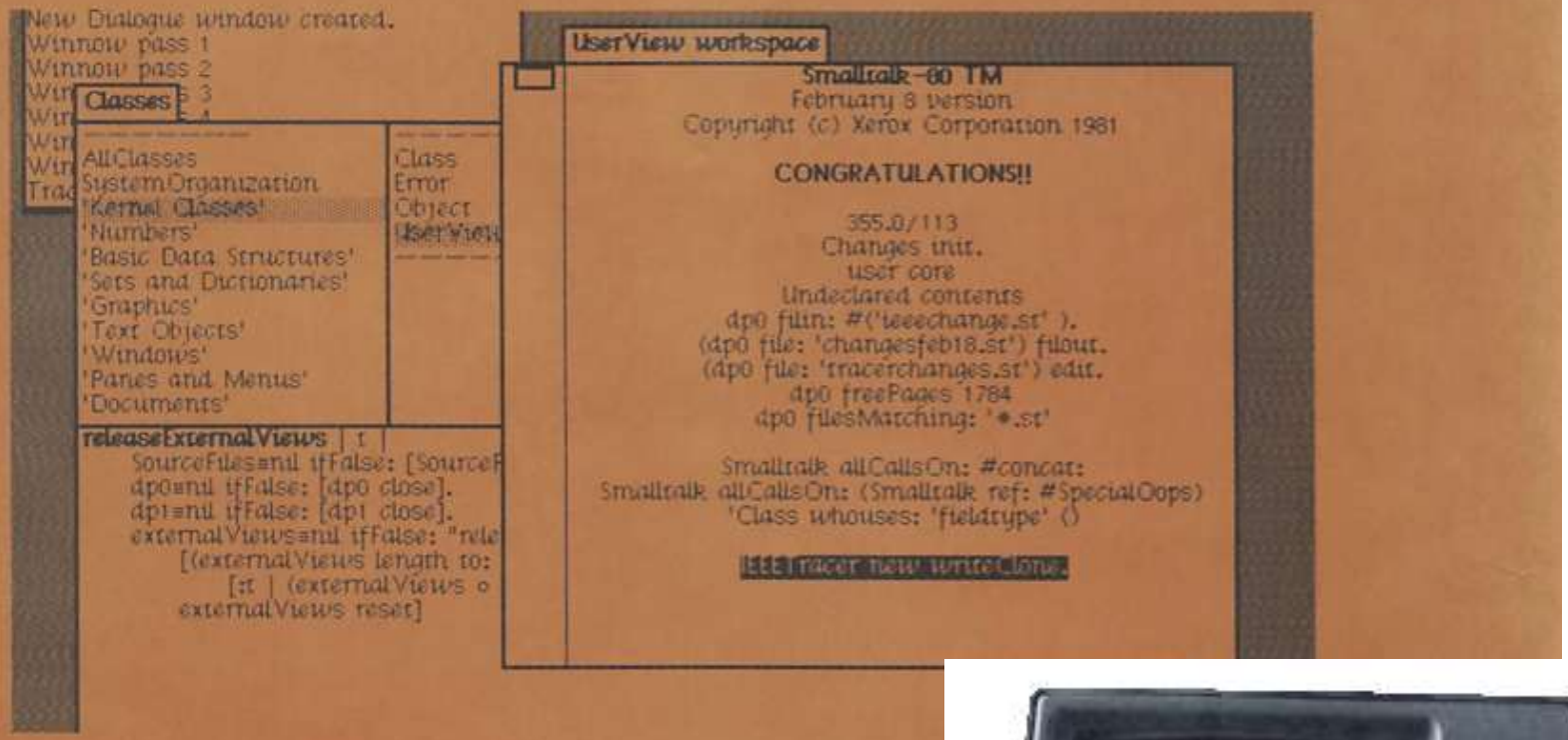


November(?) 1981 Status

http://www.wirfs-brock.com/allen/files/tek/1981-late-5-or-early-6-impl-status.pdf
http://www.wirfs-brock.com/allen/files/tek/1981-9-impl-status.pdf
http://www.wirfs-brock.com/allen/files/tek/1981-9-impl-status.pdf

July 1981

First display output of a Tektronix Smalltalk implementation.



68000 computer, virtual machine coded in GCS Pascal. RS-232 interface to a Tek 4025 raster graphics terminal.

Rendering this image took over an hour. 4025 display memory was exhausted before the complete screen could be rendered.

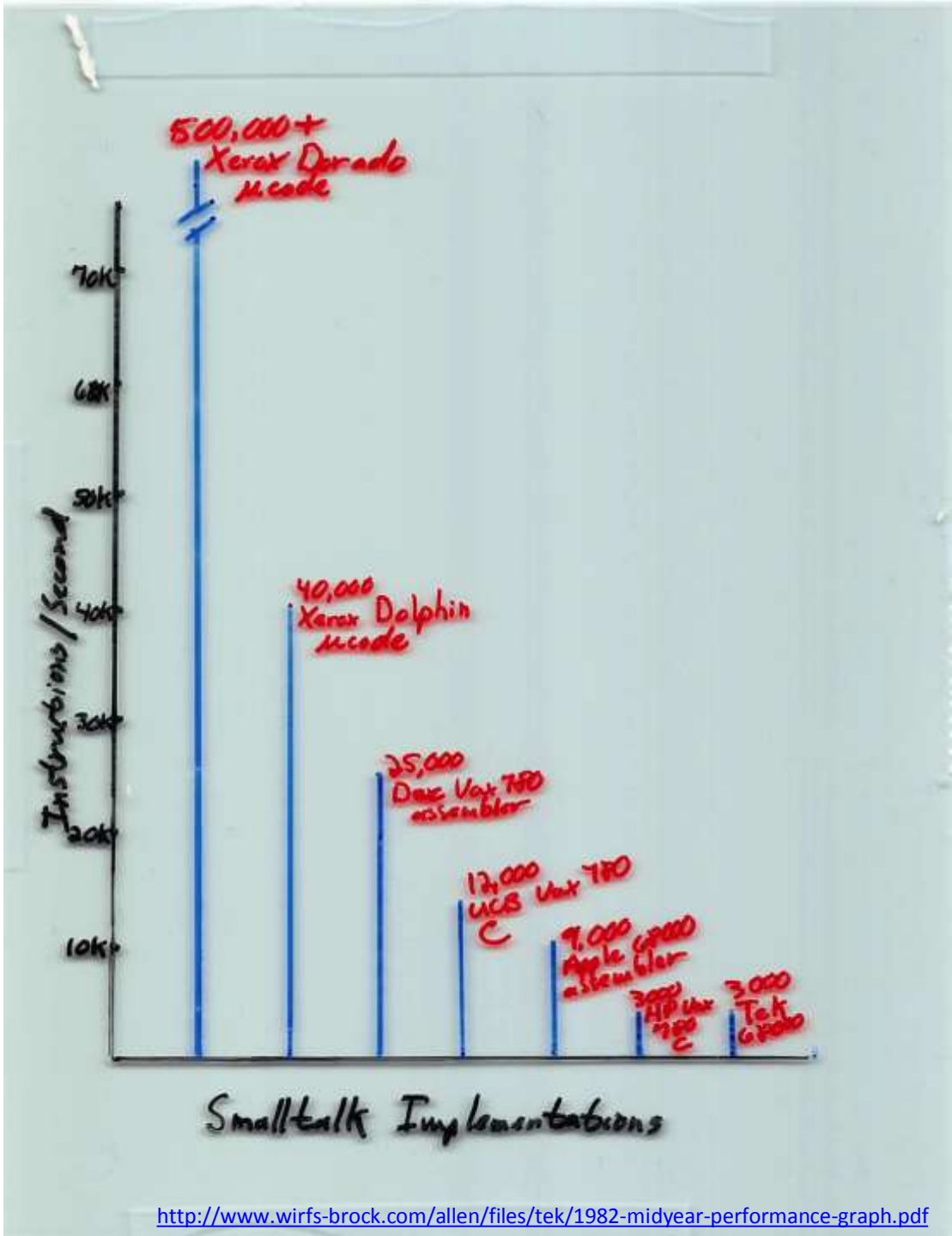
http://www.wirfs-brock.com/allen/files/tek/1981-7-first-welcome-screen.pdf

Early 1982

Performance Sucked, for Everybody Except Xerox

A reasonable interpretation of the experiment:

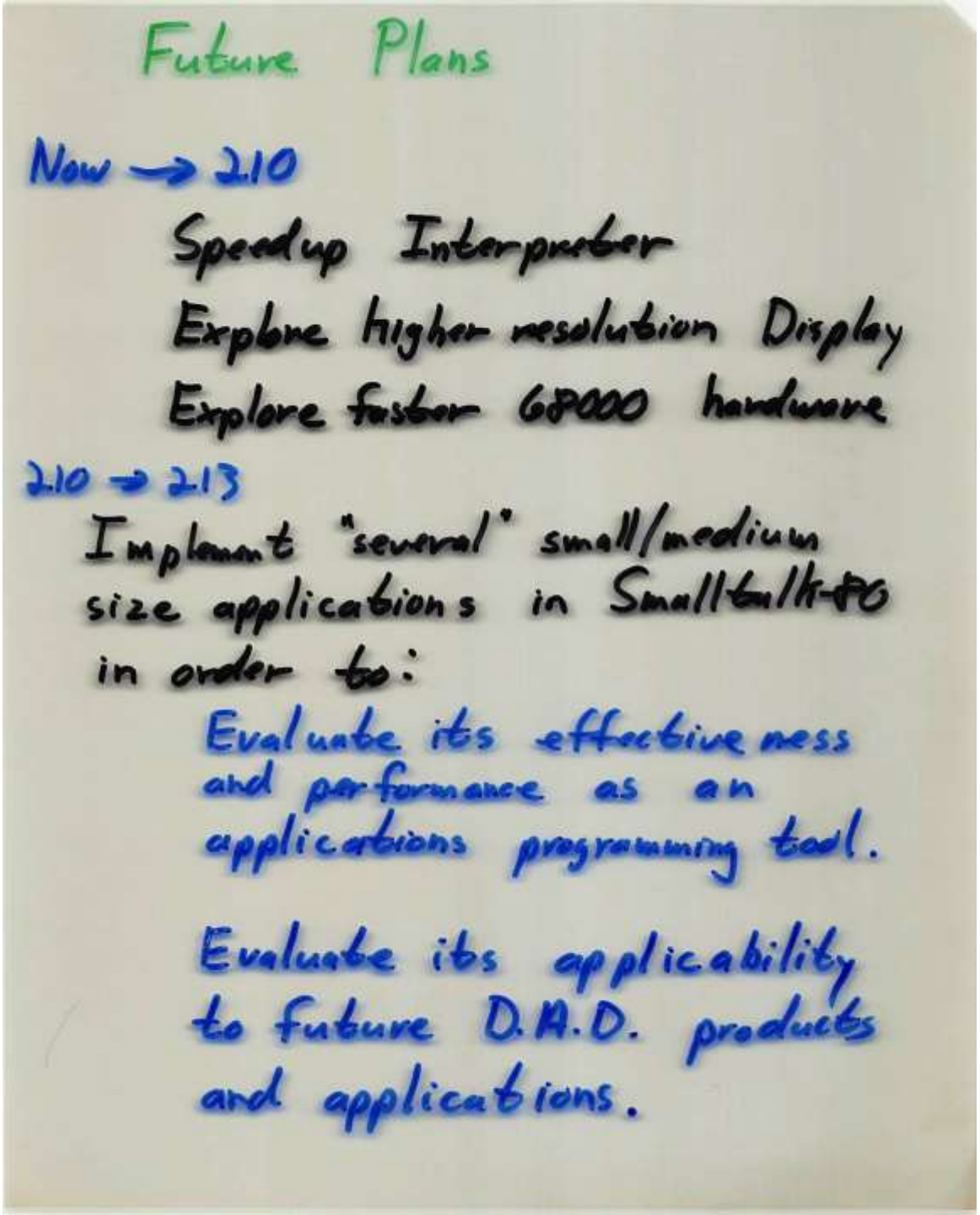
You can't usably run Smalltalk on a conventional computer.



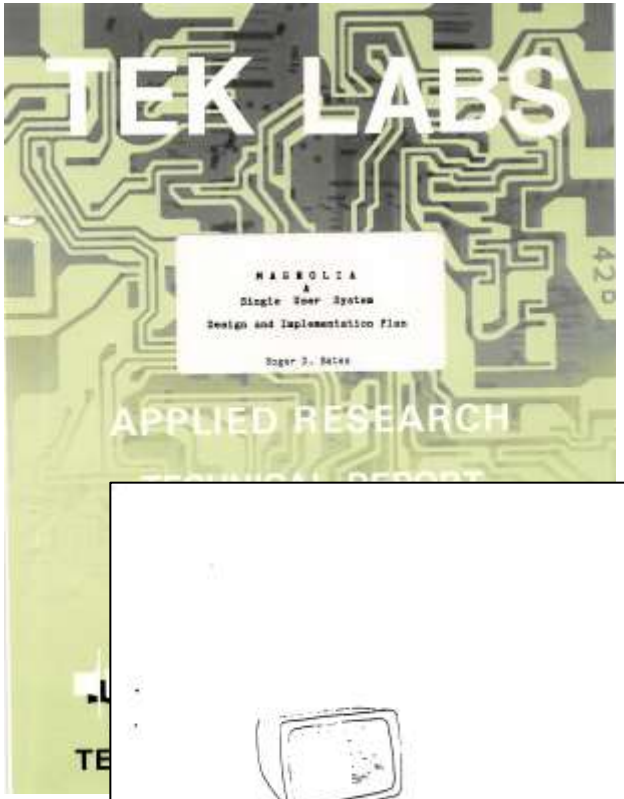
http://www.wirfs-brock.com/allen/files/tek/1982-midyear-performance-graph.pdf

Original Tek Team's Plan for First Half 1982

... When they weren't busy with their job hunt



Meanwhile at Tek Labs



MAGNOLIA
A
Single User System

Design and Implementation Plan

Roger D. Bates

Computer Research
Applied Research Group
Tektronix Laboratories

1. Introduction

The current trend of supplying "smart" terminals for use as access to central computers is an interim solution to the economic balance of computer system design. As more and more compute power is placed in the "terminal", it will be able to take over virtually all of the computational needs, and will become the computer system that the user interacts with. At this point, the display functions will become an integral part of the computing environment.

Low performance systems of this type are currently called "personal computers." In order to maintain a distinction between this type of system and higher performance systems being developed, we are calling the latter "Single User Systems."

For Tektronix to be able to move into single user systems, we need to learn how to design, build, and program this type of computer. A product that is suitable for selling to a customer will have to have not just the hardware and operating system, but a great deal of application software in order to deliver the full "capability" of the system. Tektronix will not only have to be competitive in the hardware, but in the software to be able to compete with systems of the future.

The best way to learn what to put in such a system is to provide a Single User System tool. With this we can program and use experimental tools and thereby learn what the strengths and weaknesses of this technology are. Magnolia is a tool that we can use and learn from in order to better prepare to compete in this market.

2. Design Philosophy

The motivation for this study is to present a high performance system that, if implemented, would provide a single user computing tool which would extend the resources available over that which is available on our current 11/70, or even a VAX class multi-user system. I believe that it is important that we address this class of system in order to provide for the needs of computer research in CAD, VLSI,

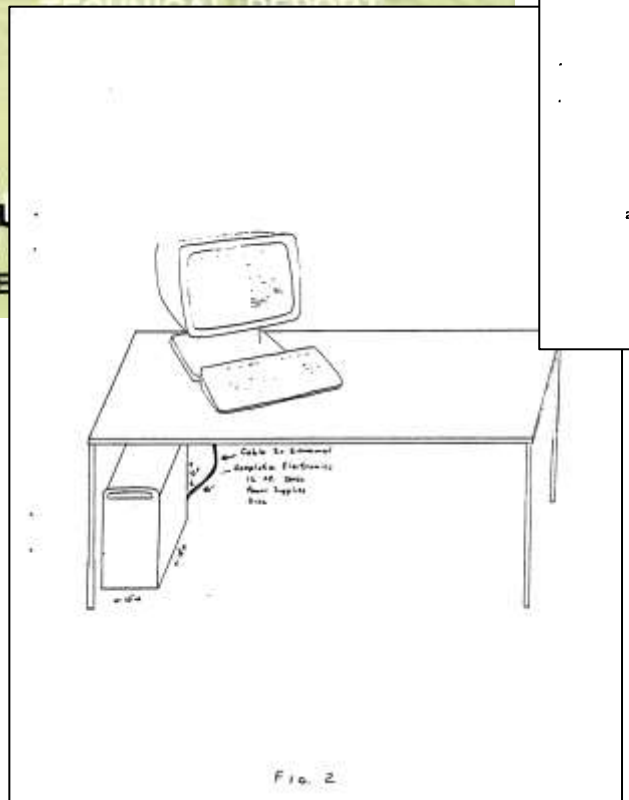


Fig. 2

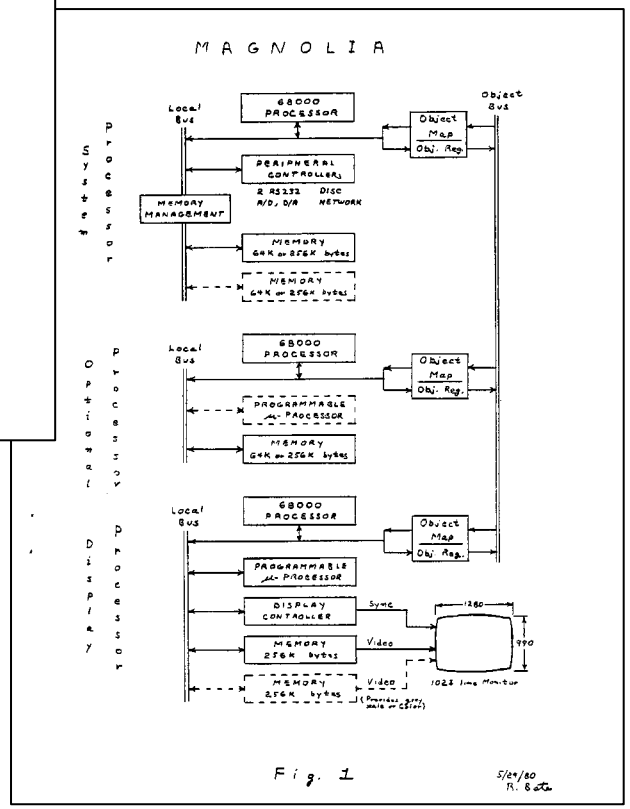
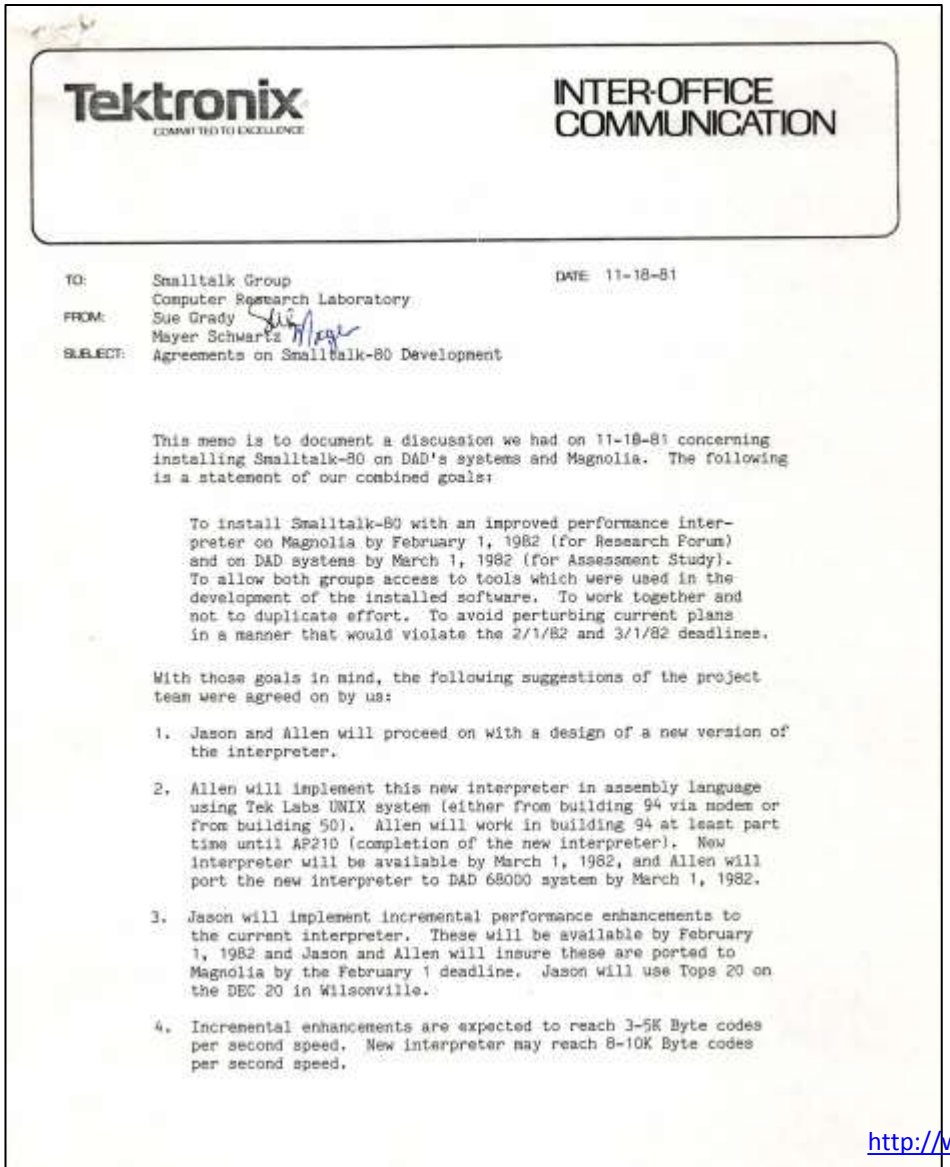


Fig. 1

5/29/80
R. B.

http://www.wirfs-brock.com/allen/files/tek/1980-9-16-magnolia-plan.pdf

Allen escapes to Tek Labs to put Smalltalk on the Magnolia



Tektronix
COMMITTED TO EXCELLENCE

INTER-OFFICE COMMUNICATION

TO: Smalltalk Group
Computer Research Laboratory
DATE: 11-18-81
FROM: Sue Grady
Mayer Schwartz
SUBJECT: Agreements on Smalltalk-80 Development

This memo is to document a discussion we had on 11-18-81 concerning installing Smalltalk-80 on DAD's systems and Magnolia. The following is a statement of our combined goals:

To install Smalltalk-80 with an improved performance interpreter on Magnolia by February 1, 1982 (for Research Forum) and on DAD systems by March 1, 1982 (for Assessment Study). To allow both groups access to tools which were used in the development of the installed software. To work together and not to duplicate effort. To avoid perturbing current plans in a manner that would violate the 2/1/82 and 3/1/82 deadlines.

With those goals in mind, the following suggestions of the project team were agreed on by us:

1. Jason and Allen will proceed on with a design of a new version of the interpreter.

2. Allen will implement this new interpreter in assembly language using Tek Labs UNIX system (either from building 94 via modem or from building 50). Allen will work in building 94 at least part time until AP21D (completion of the new interpreter). New interpreter will be available by March 1, 1982, and Allen will port the new interpreter to DAD 68000 system by March 1, 1982.

3. Jason will implement incremental performance enhancements to the current interpreter. These will be available by February 1, 1982 and Jason and Allen will insure these are ported to Magnolia by the February 1 deadline. Jason will use Tops 20 on the DEC 20 in Wilsonville.

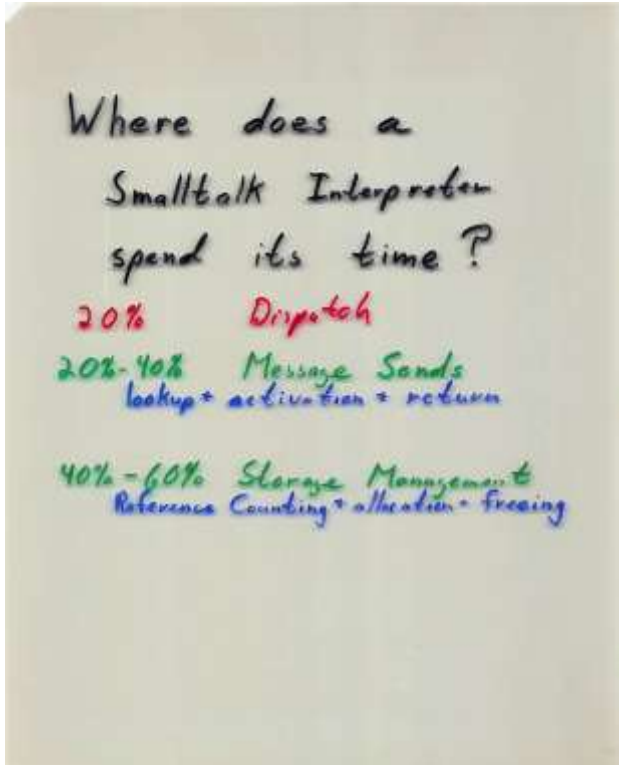
4. Incremental enhancements are expected to reach 3-5K Byte codes per second speed. New interpreter may reach 8-10K Byte codes per second speed.

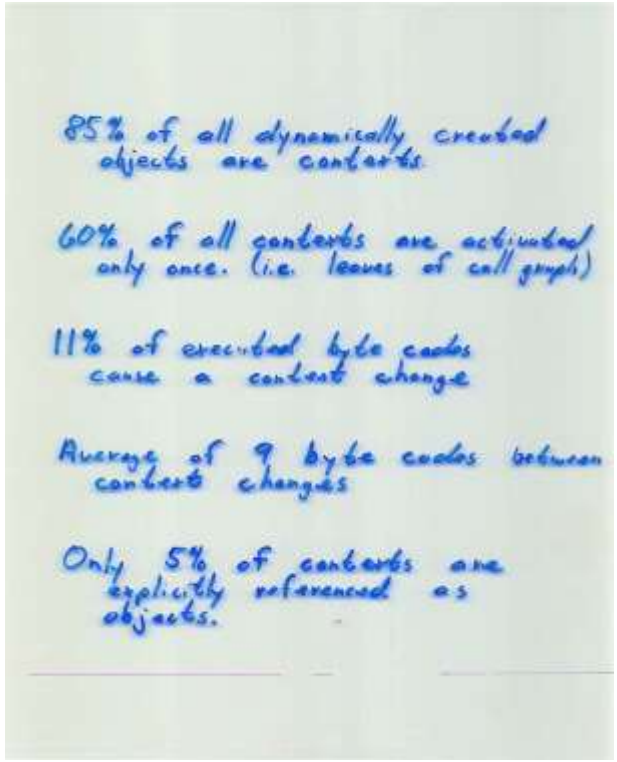
Over the next 6 months,
- Paul McCullough
- Alan Purdy
- Allen Otis
- Jason Penny

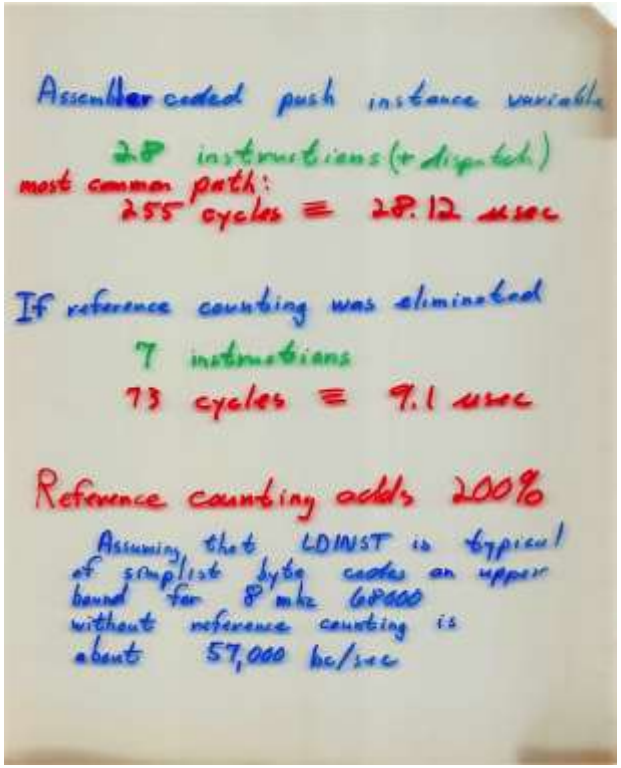
All escaped to Servio Logic and infected it with a Smalltalk vision resulting in the creation of Gemstone.

http://www.wirfs-brock.com/allen/files/tek/11-18-81-Allen-transfers-to-TekLabs-memo.pdf

Why was it so slow?

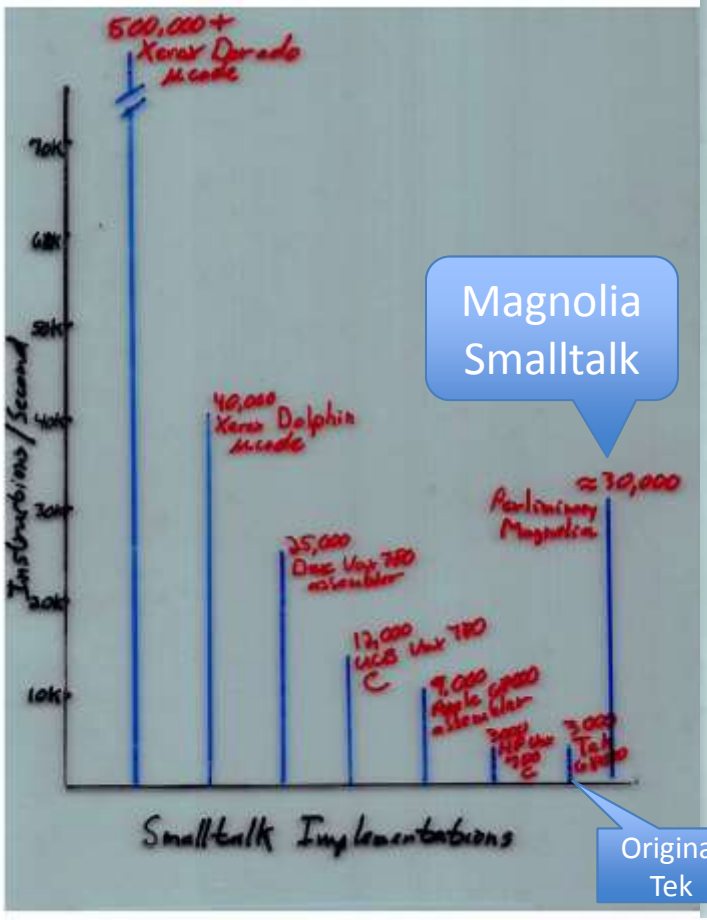


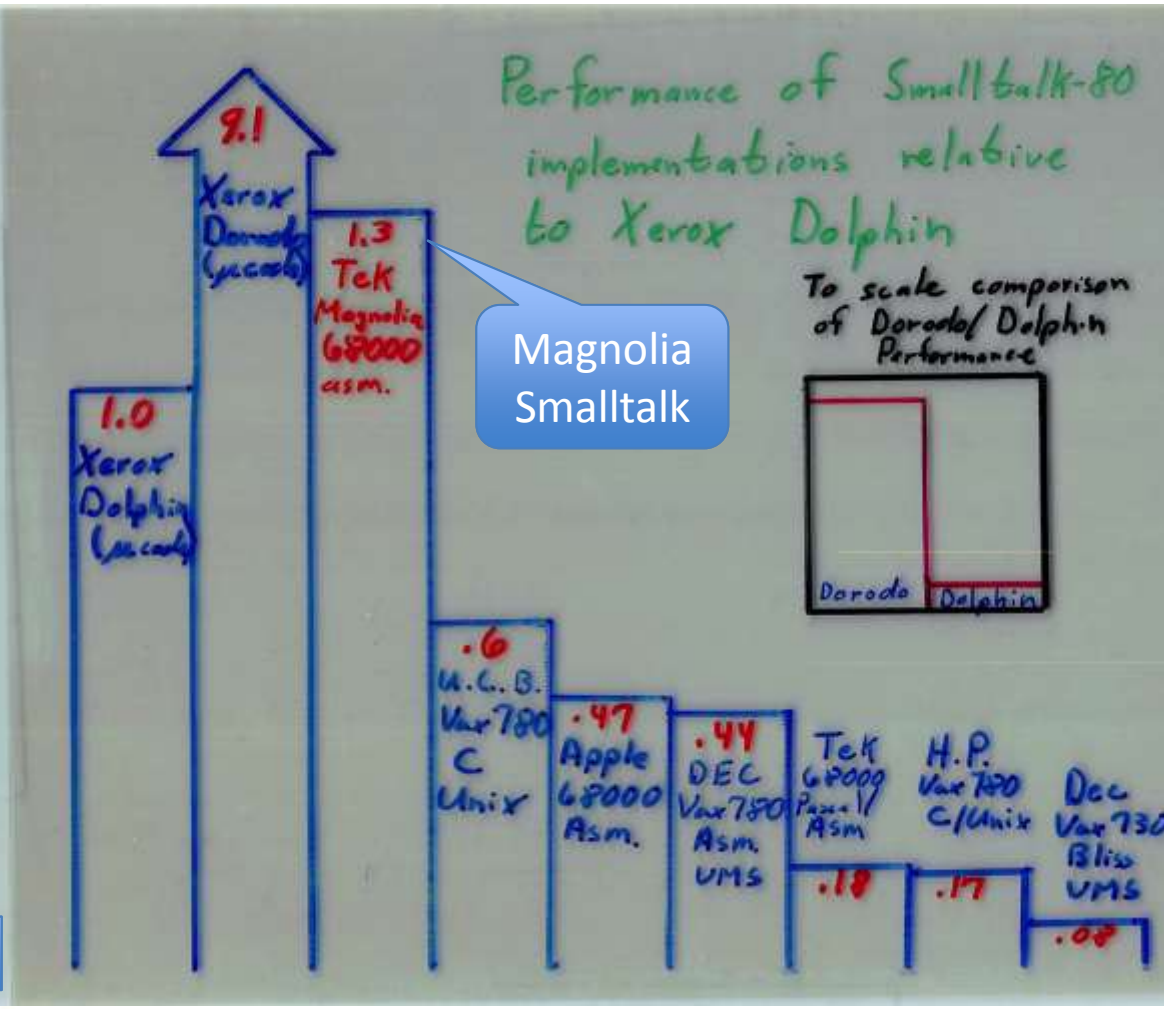




http://www.wirfs-brock.com/allen/files/tek/1983-UW-magnolia-ST-talk.pdf

It Worked!





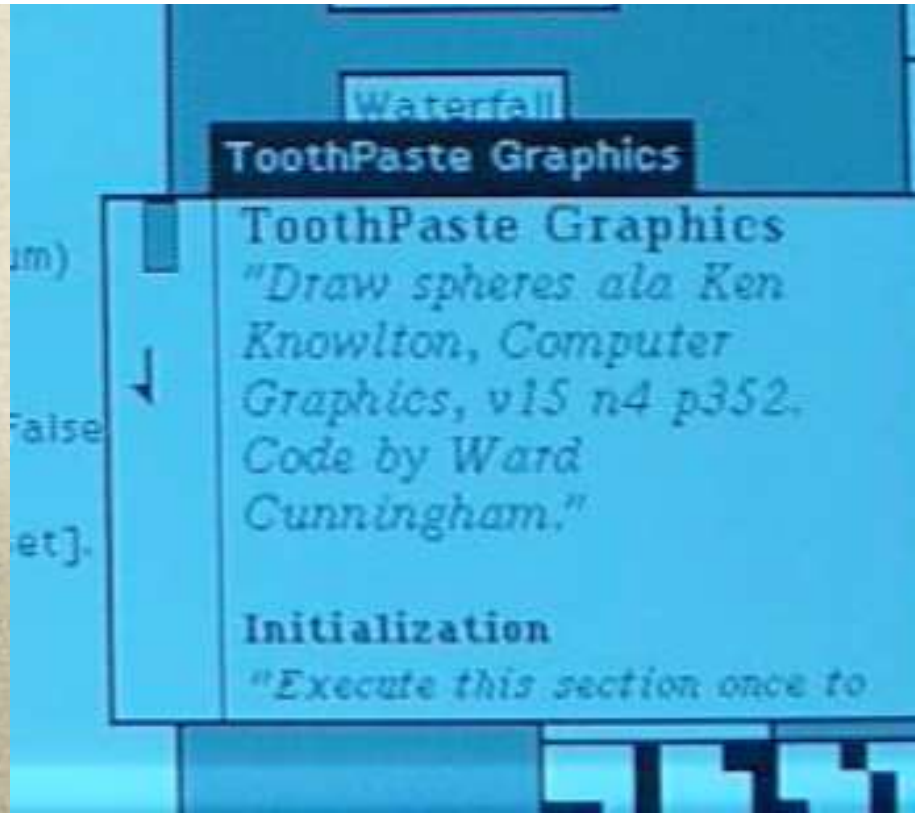
Late 1982 Performance

Mid 1983 Performance

http://www.wirfs-brock.com/allen/files/tek/1982-qtr4-magnolia-perf-graph.pdf

http://www.wirfs-brock.com/allen/files/tek/1983-Magnolia-st-perf.pdf-graph.pdf

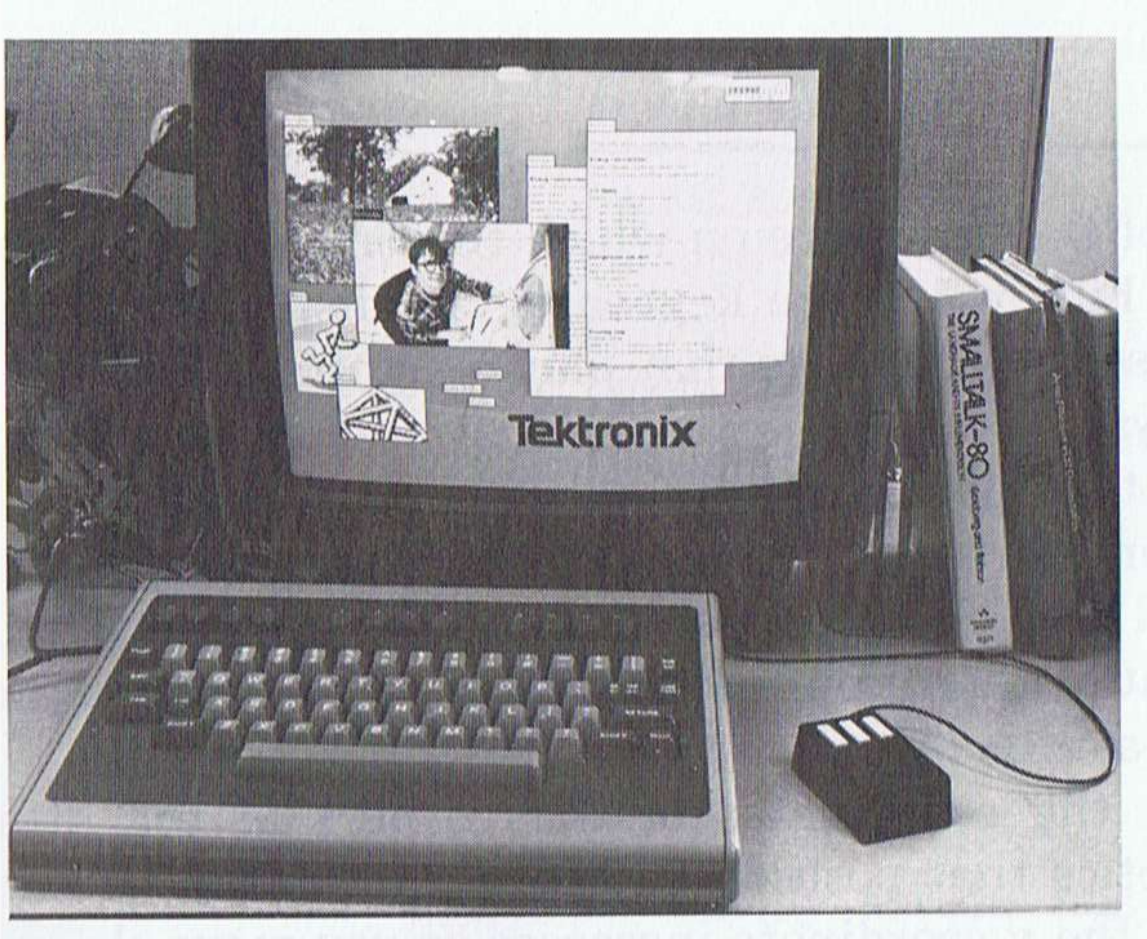
The First Tek Smalltalk User...



... And Its Most Important Evangelist

Ward Cunningham

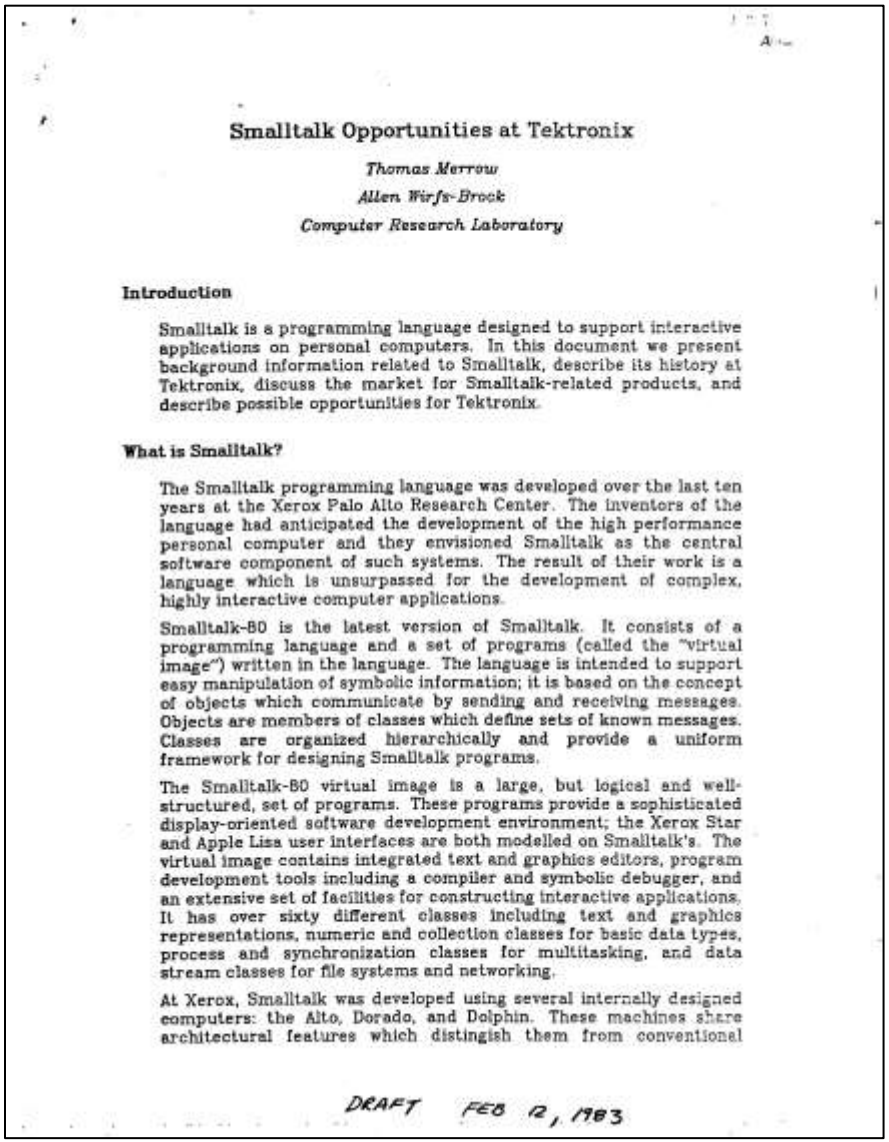
Magnolia Smalltalk rapidly became the primary language for CS researchers within Tek Labs.



In late 1982 and again in 1983 Magnolias running Smalltalk where the hit of the Tek Labs “science fair” where lab projects were show cased to the entire Tek engineering community.

Several key senior executives said: “We really should do something with this...”

Doing Something…



Smalltalk Opportunities at Tektronix

Thomas Merrow

Allen Wirfs-Brock

Computer Research Laboratory

Introduction

Smalltalk is a programming language designed to support interactive applications on personal computers. In this document we present background information related to Smalltalk, describe its history at Tektronix, discuss the market for Smalltalk-related products, and describe possible opportunities for Tektronix.

What is Smalltalk?

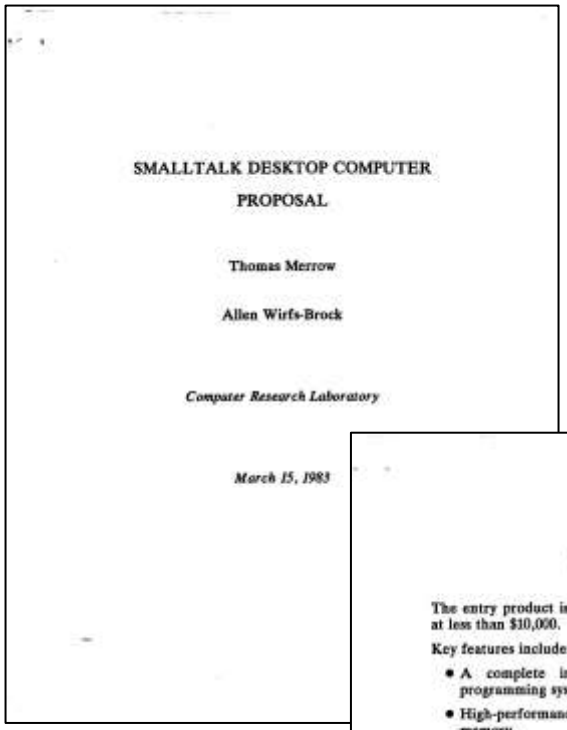
The Smalltalk programming language was developed over the last ten years at the Xerox Palo Alto Research Center. The inventors of the language had anticipated the development of the high performance personal computer and they envisioned Smalltalk as the central software component of such systems. The result of their work is a language which is unsurpassed for the development of complex, highly interactive computer applications.

Smalltalk-80 is the latest version of Smalltalk. It consists of a programming language and a set of programs (called the "virtual image") written in the language. The language is intended to support easy manipulation of symbolic information; it is based on the concept of objects which communicate by sending and receiving messages. Objects are members of classes which define sets of known messages. Classes are organized hierarchically and provide a uniform framework for designing Smalltalk programs.

The Smalltalk-80 virtual image is a large, but logical and well-structured, set of programs. These programs provide a sophisticated display-oriented software development environment; the Xerox Star and Apple Lisa user interfaces are both modelled on Smalltalk's. The virtual image contains integrated text and graphics editors, program development tools including a compiler and symbolic debugger, and an extensive set of facilities for constructing interactive applications. It has over sixty different classes including text and graphics representations, numeric and collection classes for basic data types, process and synchronization classes for multitasking, and data stream classes for file systems and networking.

At Xerox, Smalltalk was developed using several internally designed computers: the Alto, Dorado, and Dolphin. These machines share architectural features which distingish them from conventional

DRAFT FEB 12, 1983



SMALLTALK DESKTOP COMPUTER

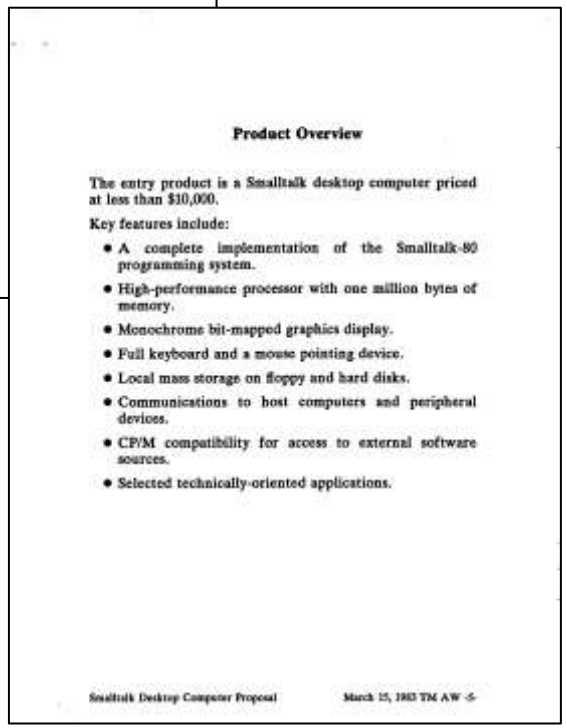
PROPOSAL

Thomas Merrow

Allen Wirfs-Brock

Computer Research Laboratory

March 15, 1983



Product Overview

The entry product is a Smalltalk desktop computer priced at less than $10,000.

Key features include:

- A complete implementation of the Smalltalk-80 programming system.
- High-performance processor with one million bytes of memory.
- Monochrome bit-mapped graphics display.
- Full keyboard and a mouse pointing device.
- Local mass storage on floppy and hard disks.
- Communications to host computers and peripheral devices.
- CP/M compatibility for access to external software sources.
- Selected technically-oriented applications.

Smalltalk Desktop Computer Proposal March 15, 1983 TM AW -5-

http://www.wirfs-brock.com/allen/files/tek/1983-2-12-Smalltalk-opportunities.pdf http://www.wirfs-brock.com/allen/files/tek/1983-3-15-Smalltalk-Desktop-Proposal.pdf

From Unicorn to Pegasus

Unicorn was a Tek IDG project (1981-82) to create a family of “low cost” color raster graphics terminals.

It produced the 4105,4107, 4109 products.

Dave Squire was the engineering manager and Rebecca Wirfs-Brock was the software project leader.

Pegasus
by
Computer Research Laboratory

10 MHz 68010 CPU
Virtual Memory Management
One megabyte of real memory
expandable to two megabytes
no wait state, 400nS
Up to 256,000 bytes of ROM storage

1024 by 1024 display bitmap
640 by 480 landscape mode monochrome display
with smooth pan over the bitmap

Uses Unicorn cabinet, power supply,
and keyboard

Small Computer Standard Interface (SCSI)
mass storage interface
Tektronix 4926 opt 25
10 megabyte Winchester disk
plus two 500,000 byte floppy disks

Mouse

Interval Timer, Clock Calendar,
Hardcopy interface, RS-232c port,
50 bytes non-volatile RAM memory,
Three voice four octave sound generator

68010-based Smalltalk machine using Unicorn packaging.

Jointly staffed by Tek Labs and Tek IDG.

Smalltalk Becomes an AI Language

Reconstructing a conversation in August or Sept. 1983 between Mike Taylor and Allen Wirfs-Brock outside Rick LeFaivre's cubicle in Tek building 50:

- Mike: We need to figure out who the Pegasus customers are going to be. So what's Smalltalk good for?
- Allen: Uh, it's a highly interactive, exploratory programming environment.
- Mike: But who needs something like that?
- Allen: (thinking about Time magazine cover he just saw about how the Japanese "Fifth Generation Computing" initiative was going to use AI to dominate the computing industry)
  Uh, It's an AI language!
- Mike: Really?
- Allen: Uh sure, it has a garbage collector and it's great for symbolic processing. It's just like LISP.
- Mike: So Pegasus is a AI machine like a Symbolics? That's incredible! That's so hot!!! We can really make a splash with that...

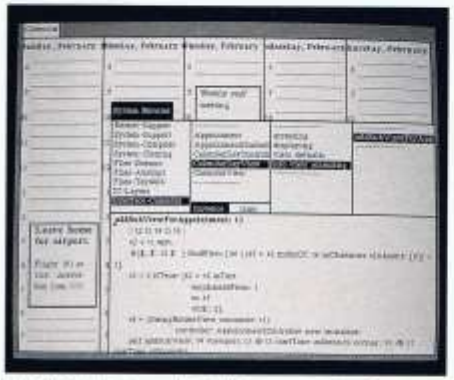
- Mike: (a week later) This AI machine thing is great! But I've done some research and it seems like people want their AI machines to run LISP and Prolog.
- Allen: Yeah, I guess we can do that too...

TEK 4404 ARTIFICIAL INTELLIGENCE SYSTEM

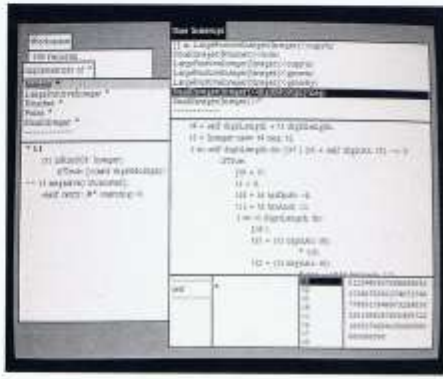
1984

THE TEK 4404: THE FIRST PERSONAL AI DEVELOPMENT SYSTEM.

AI DEVELOPMENT AT THE SPEED OF THOUGHT. AT THE PRICE OF A PERSONAL SYSTEM.

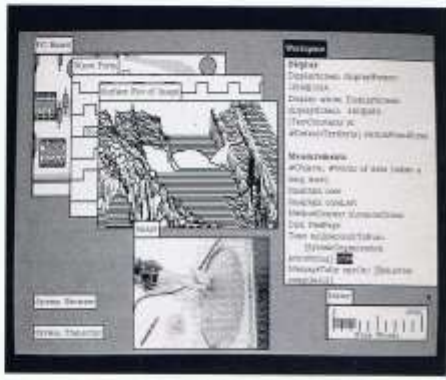


The 4404 facilitates quick, efficient prototyping. Page-on-demand memory management provides a large, 8-Mbyte virtual memory address space that permits development of complex programs without segmentation or overlays.



The bit-mapped display facilitates advanced user information concepts such as overlapping windows, "pop-up" menus, and mouse input.

The 4404's 640x480 display functions as a window into a 1024x1024 bit-map memory, with smooth panning whenever the cursor reaches a physical display edge. Users also have the ability to point with the mouse and integral joydisk.



With its proprietary Smalltalk-80 implementation, graphics performance on the 4404 makes on-screen animation possible. The Smalltalk-80 package offers advantages of a highly integrated programming environment and object-oriented language with an excellent user interface.

Tektronix
COMMITTED TO EXCELLENCE

PRICED UNDER $15,000;* THE 4404 IS AN UNMATCHED VALUE FOR AI ENVIRONMENTS.

Announced: August 6, 1984,
AAAI Conference, Austin Texas

http://www.wirfs-brock.com/allen/files/tek/4404-Flyer.pdf

A high-level programming environment suitable for artificial intelligence research, program development and delivery systems.

TEK® COMMON LISP PROGRAMMING LANGUAGE

- **A full Common Lisp implementation**
- **Optimized for the Tektronix 4400 Series of Artificial Intelligence Systems**
- **Offers a rich set of features for rapid prototyping of AI concepts**
- **Includes a resident run-time compiler for highly optimized machine code**
- **Extensive debugging information available for compiled code**

Common Lisp was conceived by a large committee of academicians and AI researchers as a language that would incorporate the very best features of other Lisp dialects. Tek Common Lisp is a full implementation of this language (as specified in "Common LISP: The Language" by Guy Steele), configured to run on the Tektronix 4400 Series of Artificial Intelligence Systems. As such, it offers a much richer set of data types and more complex program structures than other Lisp dialects currently in use.

A New Standard

Common Lisp is considered by many artificial intelligence experts to be a new industry standard for AI programming environments. The reasons for this consensus are reflected in the general parameters established for the language:

COMMONALITY: Common Lisp focuses the features of several different implementations of Lisp into a common dialect.

PORTABILITY: Applications written in Common Lisp are easily ported to any Common Lisp implementation.

EXPRESSIVENESS: Common Lisp is a very rich language that employs the most valuable constructs from other Lisp dialects.

EFFICIENCY: Common Lisp has features designed to facilitate the production of fast, high-quality compiled code.

COMPATIBILITY: Since Common Lisp is derived from a number of popular dialects, code from other Lisp dialects should readily map into Common Lisp.

Tek Common Lisp Features

Tek Common Lisp has been specifically optimized and enhanced for performance on the Tektronix 4400 Series. It provides AI researchers and software developers with a personal Lisp programming environment previously available only on dedicated Lisp machines:

- Powerful optimizing compiler with built-in debugging features
- Lexically scoped interpreter and compiler
- Full featured package system for symbol name differentiation
- Rich collection of numerical primitives and built-in functions
- Built-in garbage collector and dynamic storage management
- Complete implementation of arrays, vectors and strings
- Flexible and full-featured interactive user interface
- Powerful and flexible debugging aids
- Powerful facilities for structures and macros
- Lexical closures
- User-extensible data type facility
- Built-in user-extensible parser and hash-table facility

Tektronix
COMMITTED TO EXCELLENCE

http://www.wirfs-brock.com/allen/files/tek/85-8-commonlisp-Flyer.pdf

SONY Tektronix ●グラフィック世界のリーダー
ソニー・テクトロニクス

新製品

パーソナルでは世界初
Smalltalk™-80、LISP、Prologを一台で
オプション　オプション

Smalltalk-80

LISP　Prolog

人工知能
研究開発システム4404型

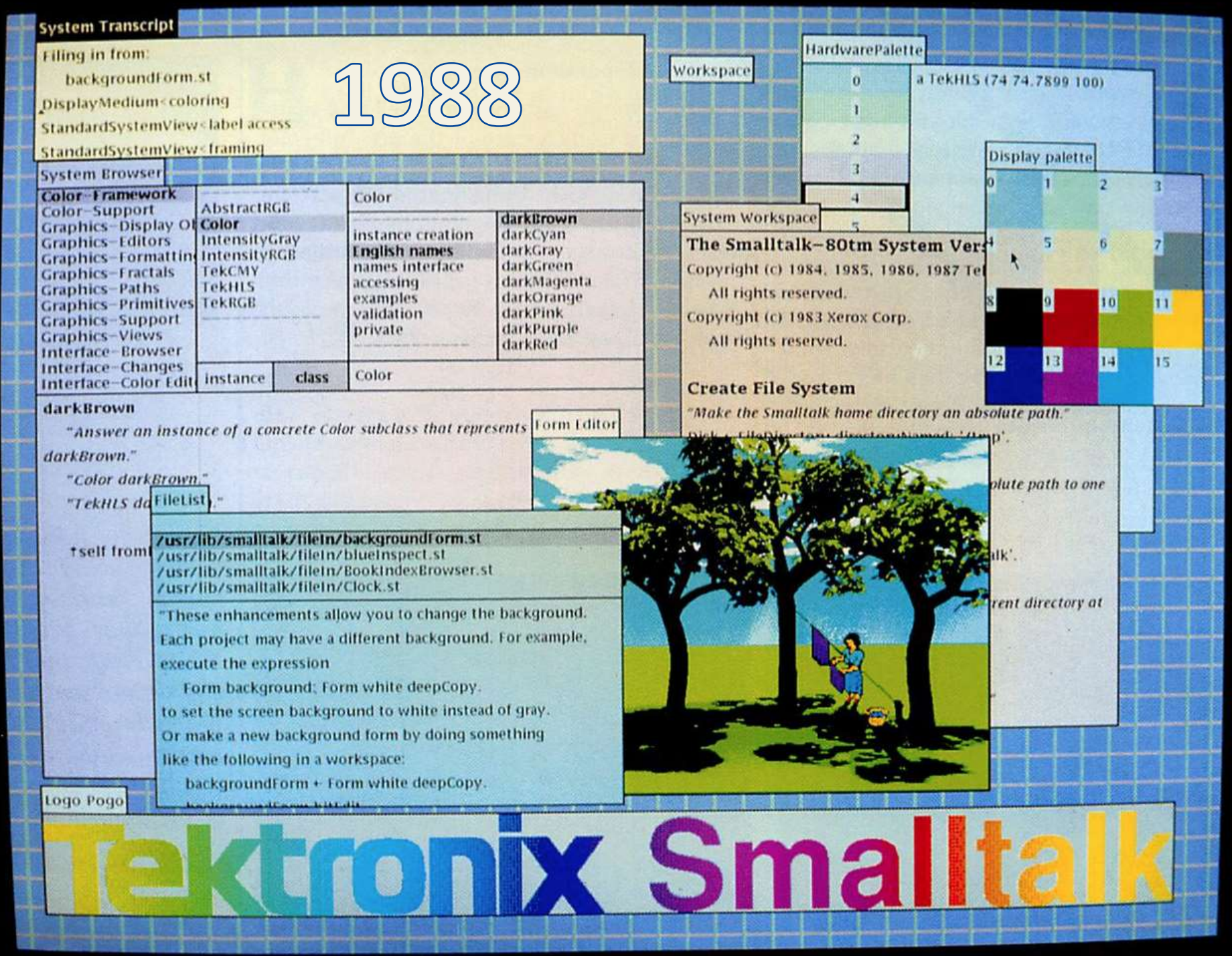
Tek LOS (Large Object Space) Smalltalk

1985-1986

Designed for 68020-based 4405 and 4406
Near Dorado Performance, 19” 1280x1024 display

- 32-bit object pointers
- No object table
- 31-bit small integers
- Multi-generation GC
- Large (>64KB) objects
- Large object GC regions
- Overlapping, stack allocated contexts
- Optimized for 68020 instructions set

We think that the Tek LOS Smalltalk may have been the first shipping commercial product, running on a off-the-self processor, to use a generational GC.


System Transcript
Filing in from:
backgroundForm.st
DisplayMedium<coloring
StandardSystemView<label access
StandardSystemView<framing
1988
Workspace
HardwarePalette
0
1
2
3
4
5
a TekHLS (74 74.7899 100)
Display palette
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
System Browser
Color-Framework
Color-Support
Graphics-Display Ob
Graphics-Editors
Graphics-Formattin
Graphics-Fractals
Graphics-Paths
Graphics-Primitives
Graphics-Support
Graphics-Views
Interface-Browser
Interface-Changes
Interface-Color Edit
AbstractRGB
Color
IntensityGray
IntensityRGB
TekCMY
TekHLS
TekRGB
Color
instance creation
English names
names interface
accessing
examples
validation
private
darkBrown
darkCyan
darkGray
darkGreen
darkMagenta
darkOrange
darkPink
darkPurple
darkRed
instance
class
Color
System Workspace
The Smalltalk-80tm System Vers
Copyright (c) 1984, 1985, 1986, 1987 Te
All rights reserved.
Copyright (c) 1983 Xerox Corp.
All rights reserved.
Create File System
"Make the Smalltalk home directory an absolute path."
darkBrown
"Answer an instance of a concrete Color subclass that represents darkBrown."
"Color darkBrown."
"TekHLS d
Form Editor
FileList
/usr/lib/smalltalk/fileIn/backgroundForm.st
/usr/lib/smalltalk/fileIn/blueInspect.st
/usr/lib/smalltalk/fileIn/BookIndexBrowser.st
/usr/lib/smalltalk/fileIn/Clock.st
"These enhancements allow you to change the background.
Each project may have a different background. For example,
execute the expression
Form background: Form white deepCopy.
to set the screen background to white instead of gray.
Or make a new background form by doing something
like the following in a workspace:
backgroundForm ← Form white deepCopy.
Logo Pogo
Tektronix Smalltalk

1988

NEW 4310 SERIES

Graphics Workstations

- **32-Bit 68020 CPU, 68881 Floating Point Processor**
- **Tektronix UTek™ Operating System**
- **Extensive Programming and Software Support, Including IBM PC Emulation**
- **X Windows with 4107 Graphics**
- **Choice of 13-inch Monochrome, 19-inch Grayscale, or 19-inch Color Displays**
- **Integrated Local Area Network**

The new Tektronix 4310 Series Graphics Workstations—the 4315, 4316 and 4317—bring together high-performance bit-mapped graphics, Tek's UTek operating system, an advanced set of programming languages, and software support for Tek 4107 and IBM PC applications. All three workstations are designed for computer-aided software engineering, 2D drafting, mapping, technical publishing, and other tasks.

Cost-Effective Performance

The 4310 Workstations are built around a 32-bit 68020 CPU and a 16.7 MHz 68881 floating point coprocessor, with a minimum of 4M bytes of RAM. They include integrated networking, an RS-232C port, and a Centronics-style parallel interface.

Each workstation includes a 1.2M byte flexible disk, an 86M byte hard disk, and a built-in SCSI to provide rapid disk input and output. With Ethernet and TCP/IP, the 4310 workstations support electronic mail, host communications, and peripheral sharing. All three workstations come with a three-button mouse and support Tek's new 4693D Color Image Printer and 4692 and 4696 Color Ink-Jet Printers.

UTek Power

Tek's UNIX-based UTek Operating System furnishes a comprehensive set of reliable utilities. Based on Berkeley 4.2bsd with a number of System V enhancements, UTek includes software development tools such as an enhanced **make** build control program, a version control system, and sophisticated queueing facilities. UTek's interactive interface simplifies system administration and maintenance.

Extensive Software

The 4310 Series Workstations come standard with a 68000 assembler, Green Hills Software's C compiler, and Tek's implementation of X Windows Version 10.4. To expand access to software, the 4310 Workstations can support Tektronix 4107 applications (via X Windows) and IBM PC/XT applications.

All three workstations support Tektronix PLOT 10® TekniCAD™ and several third-party software packages. The 4316 and 4317 support PLOT 10 TekniCAP™ also.

Smalltalk-80 and Other Advanced Languages

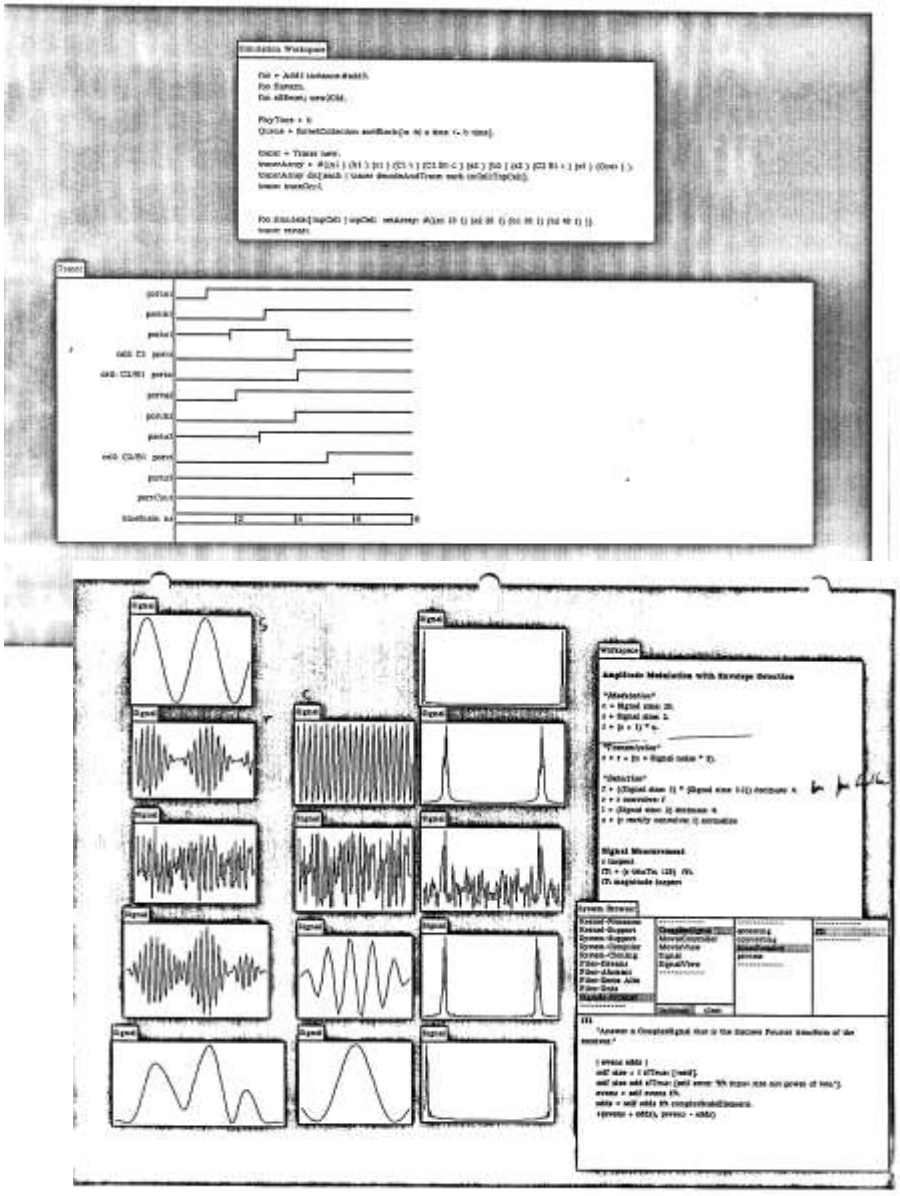
The 4310 Series provides Smalltalk-80 standard, enhanced by Tek with more than 500 additional classes of objects, an unlimited Large Object Space interpreter, grayscale and color integration into Smalltalk applications, and the ability to call functions and routines written in other languages. 4310 Workstation options include Green Hills' FORTRAN 77 and Pascal compilers, Tek Common LISP®, and QUINTUS PROLOG. Users have access to a large library of graphics routines and PLOT 10 Terminal Control System and Software Terminal Interface. Software developed for the Tektronix 4132, 4320 and 4330 Workstations is easily ported to the 4310 Series.

4310 SERIES PERFORMANCE CHARACTERISTICS

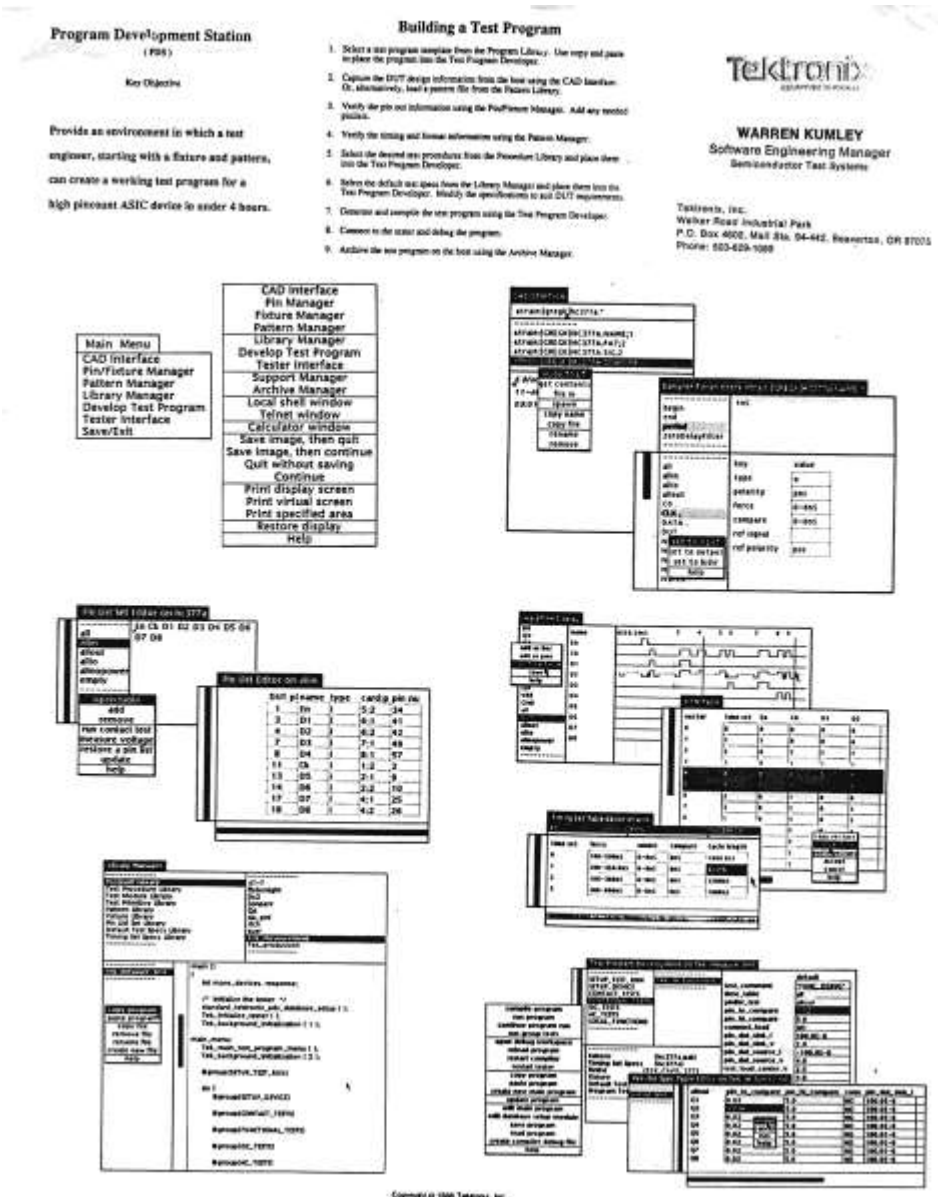
	4315	4316	4317
Processors	68020 @ 16.67 MHz 68881 coprocessor	68020 @ 16.67 MHz 68881 coprocessor Custom gate arrays	68020 @ 16.67 MHz 68881 coprocessor Custom gate arrays
Display Type	Monochrome	Grayscale (16 levels)	Color (16 from 4096)
Display Size	330 mm (13 in.)	483 mm (19 in.)	483 mm (19 in.)
Displayable Resolution	640×480	1376×1024	1376×1024
Addressability	1376×1024	1376×1024	1376×1024
System Memory (bytes)	5M	4M	4M
Optional Memory (bytes)	9 or 13M	8 or 12M	8 or 12M
Mass Storage (bytes)	86M hard disk 1.2M flexible disk	86M hard disk 1.2M flexible disk	86M hard disk 1.2M flexible disk
Optional Storage (bytes)	156 or 234M hard disk	156 or 234M hard disk	156 or 234M hard disk
Interfaces	RS-232C, LAN, SCSI, Centronics	RS-232C, LAN, SCSI, Centronics	RS-232C, LAN, SCSI, Centronics
Software	UTek, C, Smalltalk, X Windows 10.4, 4107 Virtual Terminal	UTek, C, Smalltalk, X Windows 10.4, 4107 Virtual Terminal	UTek, C, Smalltalk, X Windows 10.4, 4107 Virtual Terminal
Optional Software	Prolog, Common LISP, FORTRAN, Pascal	Prolog, Common LISP, FORTRAN, Pascal	Prolog, Common LISP, FORTRAN, Pascal
Supported Tek Printers	4692, 4696, 4693D	4692, 4696, 4693D	4692, 4696, 4693D

Applications

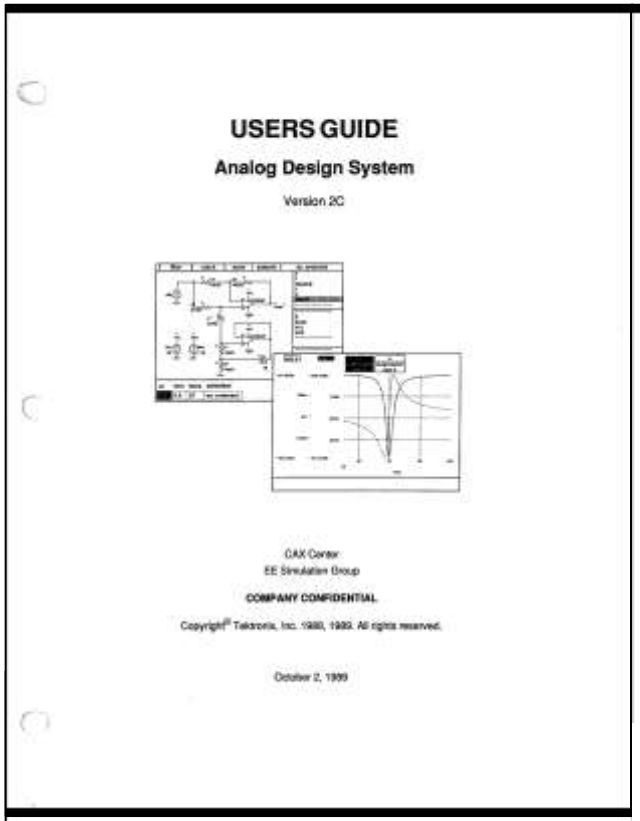
Silicon Compilers



Semiconductor Test Systems



Analog Design System



USERS GUIDE

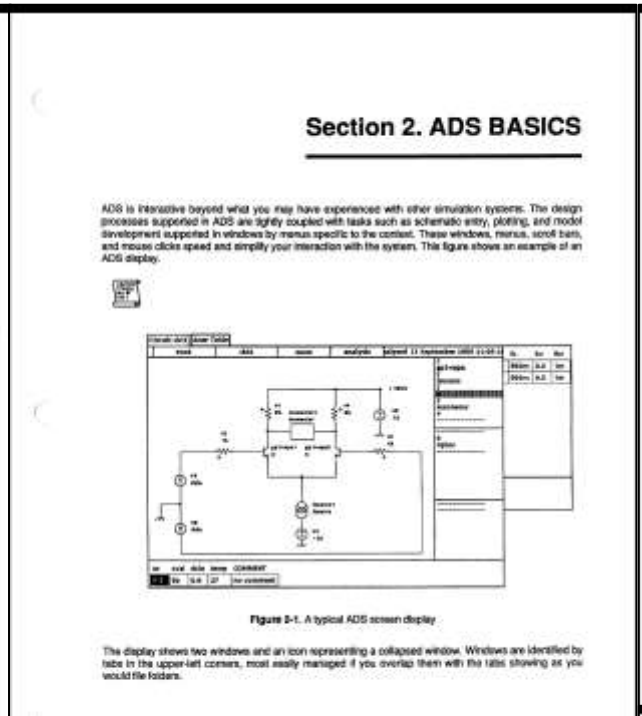
Analog Design System

Version 2C

CAX Center
EE Simulation Group

COMPANY CONFIDENTIAL

October 2, 1989



Section 2. ADS BASICS

ADS is interactive beyond what you may have experienced with other simulation systems. The design processes supported in ADS are tightly coupled with tasks such as schematic entry, plotting, and model development supported in windows by menus specific to the context. These windows, menus, scroll bars, and mouse clicks speed and simplify your interaction with the system. This figure shows an example of an ADS display.

Figure 2-1. A typical ADS screen display

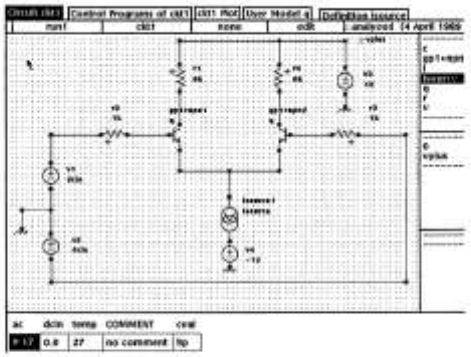
The display shows two windows and an icon representing a collapsed window. Windows are identified by tabs in the upper-left corners, most easily managed if you overlap them with the tabs showing as you would file folders.

ADS WALK THROUGH *ADS USERS GUIDE*

Using a Subcircuit

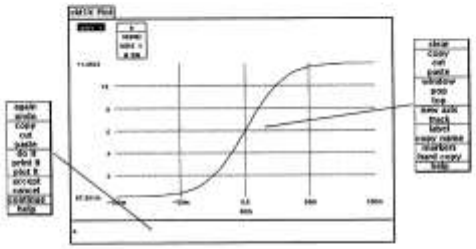
Return to the **MAIN CIRCUIT EDITOR** (in edit mode), **select** the current source and the ground symbol by dragging the cursor diagonally through a box that covers them while the left-button is depressed.

1. **cut** the selected components.
2. Click on (highlight) *isource* in the model list and **paste** the symbol in place of the current source.
3. Add a -12 volt source below the current source model. **paste** *v* from the model list and use the **ELEMENT BROWSER** to set its *dc* parameter.
4. Connect *isource* and the new voltage source into the circuit, grounding the bottom of the voltage source. (If the subcircuit and source are not connected with a wire between, the middle-button menu for a node is not available.)
5. Select the global node *vplus* and **paste** it onto the node connected to element *v3*.



ac	dcin	temp	COMMENT	cval
0.17	0.8	27	no comment	fig

Figure 30. Revised circuit with current source



PLOT Window Map

ADS stands for Analog Integrated Circuit Design System (ADS). Written in ParcPlace Smalltalk-80 (2100 classes, 33.8K methods, 12Mb source file as of 10-9-92), ADS provides a fully integrated system for drawing schematics, viewing the results of simulations, and producing design documentation. ADS has been in production use within Tektronix since June 1988. There were over 125 ADS users in 1992. The ADS program is still in production at Tektronix (as of March 2001).

Dale Henrichs started the project using the Tek 4404 in January of 1985. When Tektronix got out of the Smalltalk business he ported ADS to ObjectWorks.

ADS WALK THROUGH *ADS USERS GUIDE*

Figure 22. Selecting DC sweep parameters for changing

Click on **accept** in the middle-button menu to record the changes.

What is varied during the analysis is the variable **dcin**. While this variable is given a default value of zero in the SIMULATION CONTROL PANE below, during a dc analysis the variable is swept over the range in the increments that you specify in the dc analysis command.

3. Move the cursor to the SIMULATION CONTROL PANE (bottom) and select **execute** from the middle-button menu.
4. When the analysis finishes, return to the **SCHEMATIC EDITOR**. If in edit mode, switch to analysis mode by clicking the left-button on **edit** in the EDIT/ANALYSIS PANE, (you toggle between edit and analysis in this pane). This pane is the second one from the right at the top of the window. Point at a wire connected to the collector of one of the transistors and press the middle-button. Release the button on *v( )* and move the mouse to place the "postage stamp" plot near the collector. Click the left-button to fix the plot on the schematic.

Smalltalk-Based Oscilloscopes

From: http://www1.tek.com/forum/viewtopic.php?f=5&t=5526#p10552
Re: Console port for TDS5/7xxD oscilloscopes

Postby sschnelle on Mon Feb 11, 2013 9:45 am
Example console log i captured from my TDS794D (you can also enter commands on the console, see the 'i' command at the end):

No PCMCIA option board detected.
FLOPPY: Detected
Adding 7131 symbols for standalone.
CPU: 68EC040. Processor #0.
Memory Size: 0x1000000. BSP version 1.0.
Executing Smalltalk
-> Executing Diagnostics from Menu
Start Power-On Diag Sequence

From: http://c2.com/cgi/wiki?TektronixElevenKayScope

The TektronixInc 11xxx series of sampling oscilloscopes is an example of a successful embedded deployment of Smalltalk Language. The 11k, as it is commonly known, was a staple of the Tek product line for quite a few years.

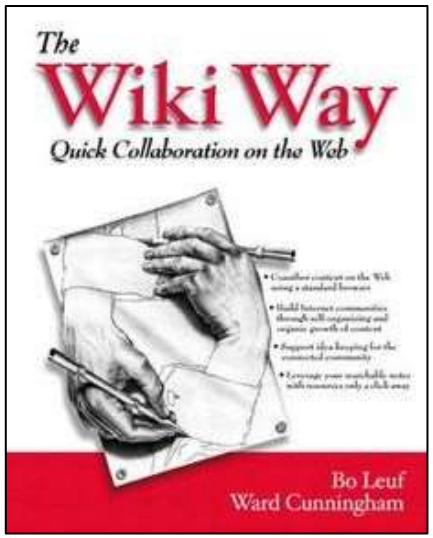
The 11k featured a 68000 processor, ample memory, and a little light on the main circuit board that turned on whenever the Garbage Collector ran. It used an embedded Smalltalk environment from OTI, and contrary to many doubters, performance was not an issue. (Unfortunately, the product abandoned the traditional UI model that oscilloscopes have, and got a reputation for being difficult to use.)

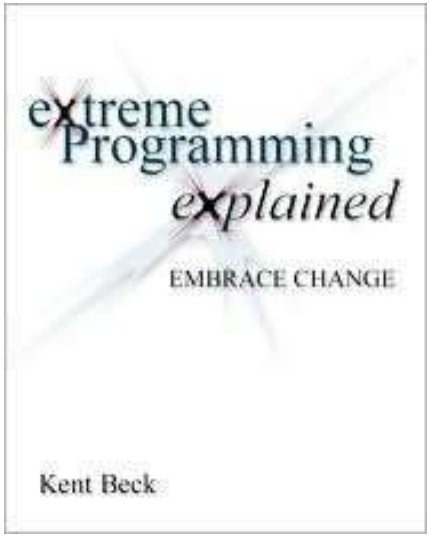
The Smalltalk environment was used in several other Tek scopes as well, the 11k was the one which survived the longest.

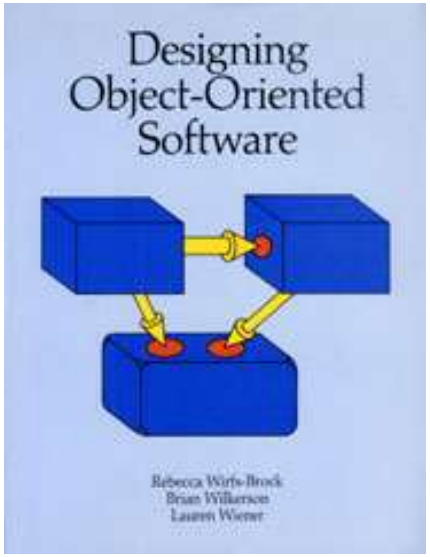
Tektronix Smalltalkers' Impact on Software Development

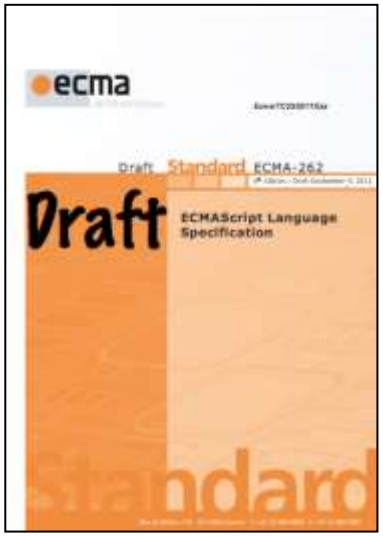
Ward Cunningham
Rebecca Wirfs-Brock
Kent Beck
Brian Wilkerson
Allen Wirfs-Brock
John Wiegand
...

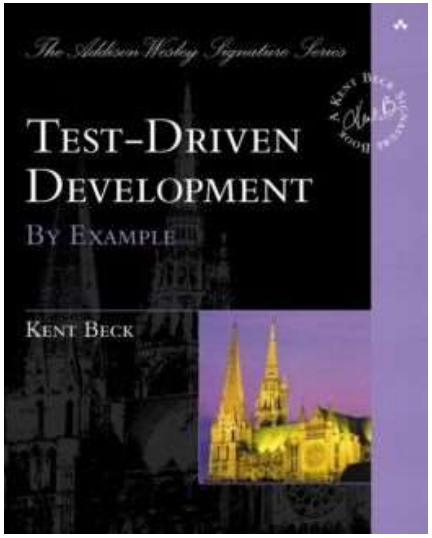
Wiki
Extreme Programming
CRC Cards
Object Behaviorism
Responsibility Driven Design
Software Design Patterns
Agile Manifesto
Pair Programming
Object Stereotypes
Test Driven Development
ANSI Smalltalk Standard
JavaScript Standards
...

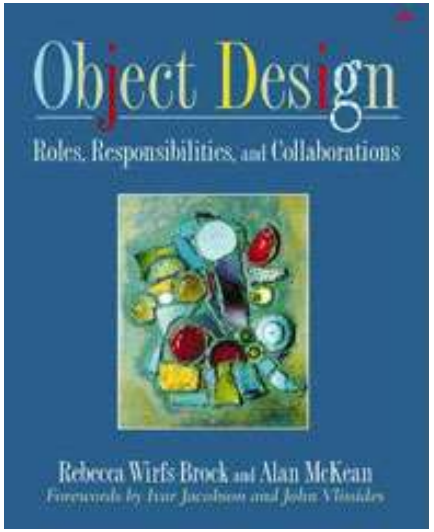


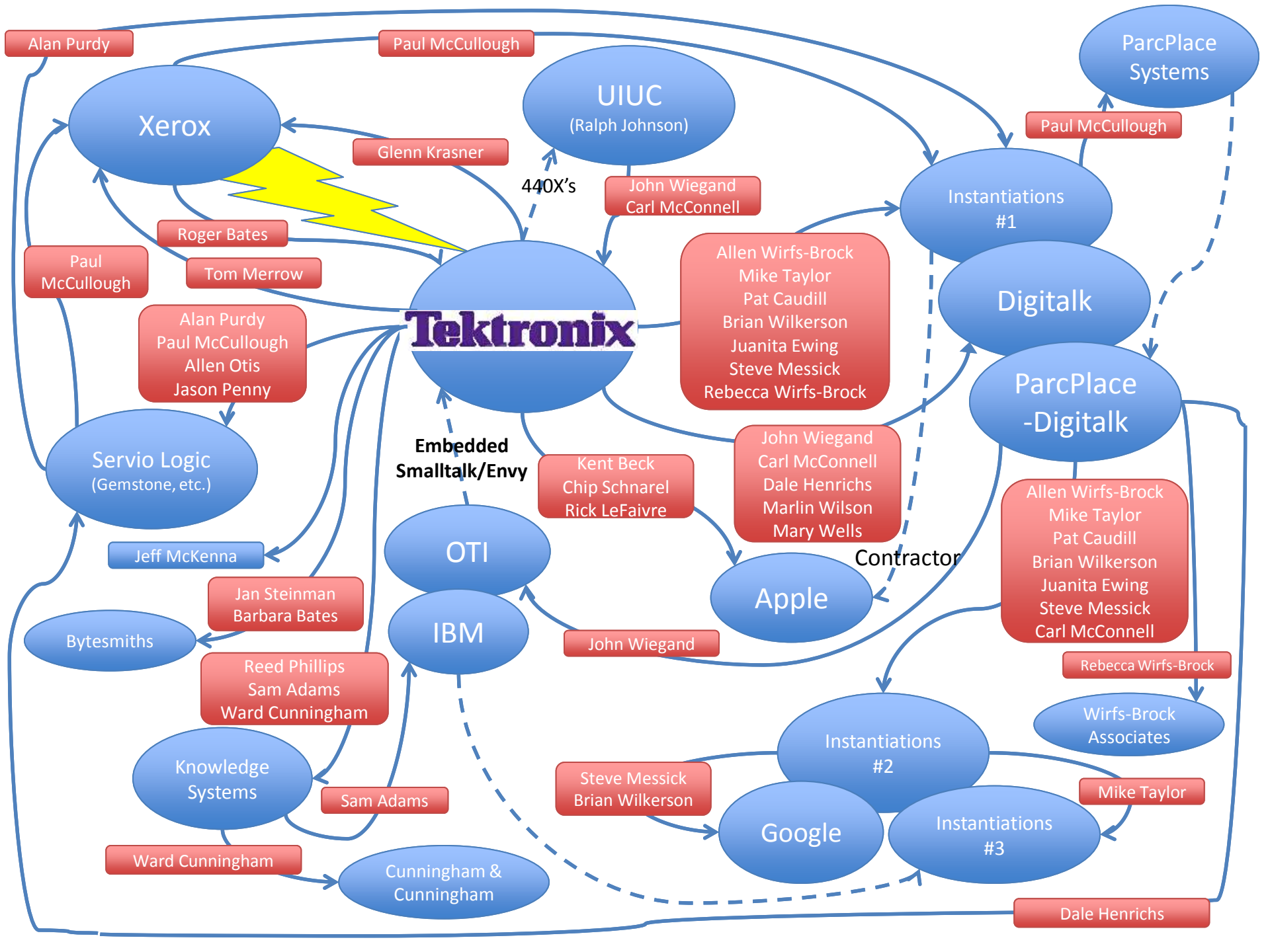











Alan Purdy
Paul McCullough
Xerox
UIUC
(Ralph Johnson)
ParcPlace
Systems
Paul McCullough
Glenn Krasner
440X's
John Wiegand
Carl McConnell
Instantiations
#1
Roger Bates
Paul
McCullough
Tom Merrow
Allen Wirfs-Brock
Mike Taylor
Pat Caudill
Brian Wilkerson
Juanita Ewing
Steve Messick
Rebecca Wirfs-Brock
Digitalk
Alan Purdy
Paul McCullough
Allen Otis
Jason Penny
Tektronix
ParcPlace
-Digitalk
Servio Logic
(Gemstone, etc.)
Embedded
Smalltalk/Envy
Kent Beck
Chip Schnarel
Rick LeFaivre
John Wiegand
Carl McConnell
Dale Henrichs
Marlin Wilson
Mary Wells
Allen Wirfs-Brock
Mike Taylor
Pat Caudill
Brian Wilkerson
Juanita Ewing
Steve Messick
Carl McConnell
Jeff McKenna
OTI
Contractor
Jan Steinman
Barbara Bates
Apple
Bytesmiths
IBM
John Wiegand
Rebecca Wirfs-Brock
Reed Phillips
Sam Adams
Ward Cunningham
Wirfs-Brock
Associates
Instantiations
#2
Knowledge
Systems
Steve Messick
Brian Wilkerson
Mike Taylor
Sam Adams
Google
Instantiations
#3
Ward Cunningham
Cunningham &
Cunningham
Dale Henrichs