



Apple Manuals Apple][Pascal



Macintosh



SwyftCard



Canon Cat



Jef Raskin Information

DOCUMENT TITLE

INFO ABOUT APPLE GUI DESIGN
FROM JEF RASKIN & BRUCE HORN

SOURCE

1996

DOCUMENT NO. 59

EX LIBRIS

DAVID T. CRAIG

736 EDGEWATER, WICHITA KS 67230 USA

E-MAIL: 71533.606@COMPUSERVE.COM



Apple Manuals Apple][Pascal



Macintosh



SwyftCard



Canon Cat

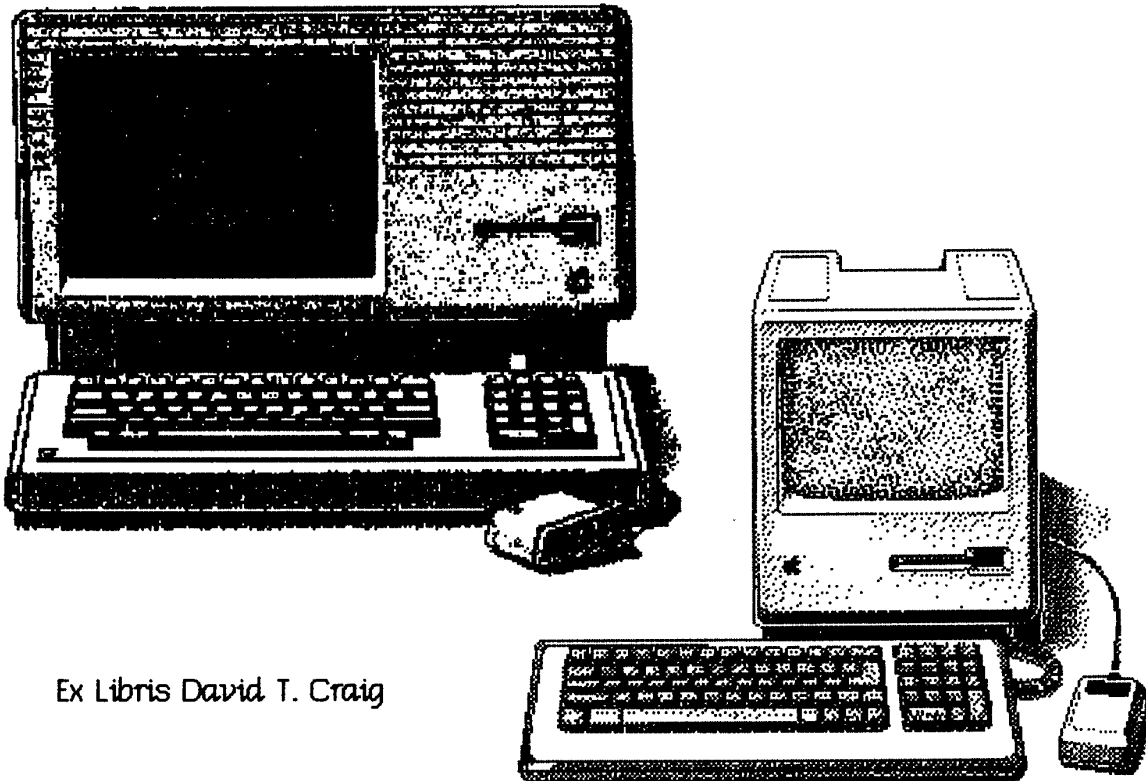


Information about Apple's GUI Interface Design from Jef Raskin and Bruce Horn

1996

Raskin began the Macintosh project at Apple and helped the Lisa team simplify the Lisa user interface (e.g. use a one-button mouse instead of a multi-button mouse).

Horn was a former Xerox Smalltalk programmer who joined Apple to create the original Macintosh Finder and define the Macintosh system software architecture.



Ex Libris David T. Craig

See p 6 for CANON CAT info

*
 * * * * *
 *
RASKIN & HORN
 * * * * *
 *

This article is from [Jef Raskin](#).

Many people have sent me copies of Bruce Horn's (attached after this message for your easy reference) comments on the sources of the design of the Mac, asking my thoughts on them. I thank you all for sending me Bruce's essay and hope you enjoy some further information.

I read Bruce Horn's recollections of the Macintosh project with some interest. Horn arrived at Apple in 1981 after the Macintosh project had been running about two years. I started working on it in early 1979, presented it to Mike Markkula about March of that year; it became an official project in September. From the chronology, it is clear why Horn doesn't have (as he is careful to say) first-hand knowledge of where many of the original ideas came from, this note can fill in a few details.

My primary role in this matter was to create the Macintosh project. I named it for my favorite kind of eatin' apple, the succulent McIntosh (I changed the spelling of the name to avoid potential conflict with McIntosh, the audio equipment manufacturer). There have been, as Horn points out, many inaccurate recountings of the early Mac history. The best I've seen (they guy did some heavy-duty research) is Owen Linzmeyer's "The Mac Bathroom Reader". My own history of the Mac (entitled "The Mac and Me") is currently being serialized in the Computer Historical Association of California's journal.

Other articles of mine that might interest readers of this: Two articles that address interface issues: * Down With GUIs! Wired, December 1993, pg. 122. * "Intuitive Equals Familiar". Communications of the ACM. 37:9 September 1994 pg. 17. Two articles that address historical matters: * "Holes in the Histories" Interactions 1.3, July 1994 pg. 11 * "Hubris of a heavyweight. A review of Steve Jobs & the NeXT Big Thing by Randall E. Stross." IEEE Spectrum, July 1994 pp. 8-9.

Horn is correct that click-and-drag methods were invented at Apple and not at PARC (or elsewhere, as far as I know). I created this method for moving objects and making selections after finding the Xerox click-move-click method prone to error. Bill Atkinson extended the paradigm to pull-down menus. This all happened relatively early in the history of the Mac. The way my insight got extended by Bill was typical of how things developed then. Surprising as it may seem in retrospect, there was some resistance to my new way of using a graphic input device and I had to repeatedly explain how drag worked and why it was often easier to use than the modal click-move-click technique developed first (as far as I know) on the Sketchpad system and then used at Xerox PARC. Some of the arguments I used involved looking at number of user actions and the time they took, an approach that was then or would soon become the very useful GOMS model of Card, Moran, and Newell. Bill was a strong supporter of my ideas and at one session where I was explaining how drag worked Bill, by way of amplifying how useful it was, said something like, "And you can use it to open menus, just put the cursor on the top and drag down to the item you want."

I hired Bill for Apple, inviting him up from UCSD, where he had been a student of mine. His close friend Bud Tribble, another UCSD student I knew, joined us. Later still Bud was to lead software development at Next.

Trying to untangle the history is sometimes hard, as in my reference to the work of Card et. al. To see what I mean, here's a bit of background: I had been, in the early 70's, a professor and computer center director at the University of California at San Diego and a Visiting Scholar at the Stanford Artificial Intelligence Laboratory (SAIL) at the now-demolished D.C. Power Laboratory (named for a Mr. D. Power). When PARC was in its first few years I was often a visiting academic there, taking part in discussions and viewing with delight some of the developments going on there; I trust that people there also took pleasure in finding in me someone who was already on much the same user-interface wavelength. I didn't have to be sold on the idea that UI and graphics were of primary importance to the future of computing. When I joined Apple in 1978 I stopped visiting PARC to avoid any possible conflicts of interest. Given these circumstances I could have learned from Stu Card, Tom Moran or others at PARC the basic ideas of GOMS style analysis, or I might not have run into that work during my visits to PARC. I don't remember, and unless I find something in my files about it someday or someone else recalls a significant event, I will never know if my primitive GOMS-style analysis that helped lead to Apple's adopting my click-and-drag methods was based on their work or not.

I was the 31st employee at Apple (joining in January, 1978), but I had first met Jobs and Wozniak in their garage in 1976, and told them of the wonderful work being done at PARC. Working on the Apple I at the time, they weren't interested in human factors. While I was the first PARC-savvy person at Apple, Larry Tesler was the first PARC employee to join the company. At first he was strongly opposed to the Mac's easier-to-use mouse methods, and I eventually wrote a memo that showed, point by point, that the one-button mouse could do everything that PARC's three-button mouse could do and with the same number or fewer user actions. It was faster and more efficient, and much easier to learn and remember how to use. I had observed that people (including myself) at PARC often made wrong-button errors in using the mouse, which was part of my impetus for doing better.

Horn makes it seem that the selection-based editor came with Tesler from PARC. It may have been a case of convergent evolution, since we already had that paradigm at the Mac project. In this case it dates at least back to an editor I designed much earlier, while at Bannister & Crun. In '73 I discussed my editor concepts with many people at PARC, so I do not know whether Tesler's design was influenced by my work, I know it was not the other way around.

My thesis in Computer Science, published in 1967, argued that computers should be all-graphic, that we should eliminate character generators and create characters graphically and in various fonts, that what you see on the screen should be what you get, and that the human interface was more important than mere considerations of algorithmic efficiency and compactness. This was heretical in 1967, half a decade before PARC started. Many of the basic principles of the Mac were firmly entrenched in my psyche. By the way, the name of my thesis was the "Quick-Draw Graphics System", which became the name of (and part of the inspiration for) Atkinson's graphics package for the Mac.

Thus Horn is more correct than he knew when he wrote that the world has generally overestimated the influence of PARC on the Mac, as even some of the concepts that he attributes to PARC's influence predated PARC.

When Bruce Horn discusses the hardware, and attributes the overall concept for the design to Jerry Mannoek (who indeed did a world-class job on the final design of the box). Horn is unaware that the requirements for a small footprint, unique aesthetic, built-in sound, etc. were all part of the project specs from the very beginning, long before Mannoek joined the team. I still have drawings (mostly by my long-time friend and fellow early Apple employee Brian Howard) starting in late 1979 that show these features. The built-in speaker and serial ports were merely carrying on the Apple II tradition. My design also had a bus extension port to allow additional memory and other devices to be added, but Steve Jobs removed that feature. It returned with the SE.

Before creating the Mac project, I was Manager of Publications at Apple, and so for the Mac I was careful to insist that the excellence of the product extend, to use Horn's words, to "the unpacking instructions, the profusely-illustrated and beautifully-written manuals, ... tastefully packaged." Packaging was another major concern of mine; I had worked for a company in South San Francisco called "The Box Factory" where I had done box and display design. I did not work on the manuals or the box design myself, but I had put in place the systems and people who would do a first-rate job, and inculcated these values in Apple's management.

Horn and many other people who joined Apple long after the Mac and Lisa projects were well under way never knew the genesis of many of the ideas that were later to become prominent and widely copied. For example, he attributes the internationalization of the products to the Lisa group, but it actually began when I hired Joanna Hoffmann into the Mac group partially because of her international background and my interest in providing international fonts. I could go on for another few pages of similar small errors in chronology or attribution in Horn's remarks, but they are not of major importance (except perhaps to the people who did the work).

Time plays tricks on memory. By chance, I got to use a Xerox Star for the first time recently. It was in a room with a Lisa and an early Mac. I found the Star and the Lisa to be incredibly slow and somewhat clumsy to use; the Mac was far faster and more fluid (a tribute to Horn and his associate's efforts!). The speed differences were real, but it could be that my familiarity with the Mac accounts for the feeling of clumsiness with the Star and Lisa. I suspect if Horn were to go and use a Star today, he would not be quite so enthusiastic about how "advanced" it was, at least from a user's point of view. Of course, he might well be correct when speaking from a programmer's point of view. However, I've always been more concerned with users. Programmers do their work but once, while users are saddled with it ever thereafter.

I find looking at the past not nearly as interesting as looking into the future. As Horn says, and as I have written in a number of articles, things haven't changed as much as the hype would have it. I think that years from now, when the details have been washed away by the acid rains of time, four major commercial events will stand out in the history of personal computers: the advent of the microprocessor which drove prices of computers down to the point where individuals could buy them and led to the first flowering of the present computer revolution, the ascendancy of the software industry and the shift from "users will program them" to "users will run software packages", the Mac interface and its followers which brought the benefits of computers to a far broader audience and fundamentally changed the way we use computers of all sizes and software of all kinds, and (to tread on dangerous ground since the event is relatively recent) the blossoming of the Internet. To sum up the history:

- Cheap Hardware
- Application Software
- Human Interface
- Internet

Forget operating systems as a significant part of the story, they are just a detail. Users would be better off if they never had to deal with one, and someday operating systems will disappear from view, just as the details of the processor (thankfully) have.

The future lies with getting rid of the cumbersome and complex systems we have now and moving to simpler, more direct methods of harnessing the power of the processor. See my article in *Wired* (cited above) for a bit more on the reasons for this.

This tidbit is from Bruce Horn. Bruce is one of the original Macintosh Division engineers. He co-authored the Finder. (A short bio is at the end.)

On Xerox, Apple, and Progress

I've been watching the debate for more than a decade now about where the Macintosh User Interface came from. Most people assume that it was taken directly from Xerox, after Steve Jobs went to visit. This "fact" is reported over and over, by people who don't know any better (and also by people who should!). This just isn't true--there are some similarities between the Apple interface and the various interfaces on Xerox systems, but the differences are substantial.

Steve did see Smalltalk when he visited PARC. He saw the Smalltalk integrated programming environment, with the mouse selecting text, popup menus, windows, and so on. The Lisa group at Apple built a system based on their own ideas combined with what they could remember from seeing the Smalltalk demo, and the Mac folks yet another. There is a significant difference between using the Mac and Smalltalk (or the Star; I don't know if Steve ever saw that, and none of the Mac group did before the Mac came out).

There was no Finder in Smalltalk, and no need for one, really. Drag-and-drop file manipulation came from the Mac group (the Finder prototype was shown to the Lisa group, which implemented their own version), along with many other unique concepts: resources and dual-fork files for storing layout and international information apart from code; definition procedures; drag-and-drop system extension and configuration; types/creators for files; direct manipulation editing of document, disk, and application names; redundant typed data for the clipboard; multiple views of the file system; desk accessories; and control panels, among others. The Lisa group invented quite a few fundamental concepts as well: pull down menus, the imaging and windowing models based on QuickDraw, the clipboard, and cleanly internationalizable software. Smalltalk had a 3-button mouse and popup menus, in contrast to the Mac's menubar and one-button mouse. Smalltalk didn't even have self-repairing windows--you had to click in them to get them to repaint, and you couldn't draw into windows that were partially obscured. Bill Atkinson did not know this, so he invented regions as the basis of QuickDraw and the Window Manager so that he could draw in covered windows and repaint portions of windows brought to the front easily and quickly. The one thing that I can recall where a Macintosh feature is identical to one in Smalltalk is the selection-based modeless text editing with cut and paste, which was created by Larry Tesler for his Gypsy editor at PARC.

The difference between the Xerox system architectures and the Mac's architecture is huge; much bigger than the difference between the Mac's architecture and Windows. Not surprising, since Microsoft was shown quite a bit of the Macintosh design (API's, sample code, etc.), during the Mac's development from 1981-1984; the intention was to help their programmers write applications for the Mac, but having this also gave their system designers a template from which to start designing Windows. In contrast, the Mac and Lisa designers had to invent their own architectures. Of course, there were some ex-Xerox people in the Lisa and Mac groups, but the design point for these machines was so different that we didn't leverage off our knowledge of the Xerox systems as much as some people think.

The hardware itself was an amazing step forward as well. All-in-one design, 4-voice sound, small footprint, clock, auto-eject floppies, serial ports, and so on. The small, portable, appealing case was a serious departure from the ugly-box-on-an-ugly-box PC world, thanks to Jerry Manock and his crew. Even the packaging showed amazing creativity and passion; do any of you remember unpacking an original 128K Mac? The Mac, the unpacking instructions, the profusely-illustrated and beautifully-written manuals, and the animated practice program with audio cassette were packaged together tastefully in a cardboard box with Picasso-style graphics on the side. Never before had a computer been delivered with so much attention to detail and the customer's needs.

Read Guy Kawasaki's books, "The Macintosh Way" and "Selling the Dream," and you'll get an idea about what **really** went on at Apple. In particular, "Selling the Dream" includes the entire Macintosh Product Introduction Plan in an appendix. The Mac project was innovative from top to bottom, from engineering to marketing to sales to evangelism to product design. We can thank Steve Jobs for hiring the kind of people who brought the enthusiasm, creativity, and passion for excellence that made the Mac possible.

Looking Back

In my opinion, the software architectures developed at Xerox for Smalltalk and the Star were significantly more advanced than either the Mac or Windows. From what I have read since, the Star was a tremendous accomplishment, with features that current systems haven't even started to implement; OpenDoc will be a strong advance past the Xerox systems. I have great respect for the amazing computer scientists at Xerox PARC, who led the way with the innovations that we all take for granted now, and from whom I learned a tremendous amount about software design. Still, the Lisa and Macintosh groups deserve respect for making a large number of these innovations available to personal computer users, and a significant number of homegrown innovations as well.

Apple could have developed a more complex, sophisticated system rivaling the Xerox architectures. But the Mac had to ship, and it had to be relatively inexpensive--we couldn't afford the time or expense of the "best possible" design at every step of the way. As a "little brother" to the Lisa, we didn't have multitasking or protection, since we didn't have space for the extra code or stack required. Our memory and disk constraints were very tight: for example, the Resource Manager took up less than 3K bytes of code in the ROM, and the Finder 46K on disk. There were **many** design decisions that we regretted to some extent--even at the time some of us were disappointed at the compromises we had to make--but if we had done it differently, would we have shipped at all?

The Past and Future

In many ways, the computing world has made remarkably small advances since 1976. I suspect that some of the Star features still are not available on any platform. We continually reinvent the wheel: Smalltalk had a nice bytecoded multiplatform virtual machine long before Java. As far as language technology is concerned, Object-Oriented Programming is the hot thing now, and it's almost 30 years old (see Simula-67). Environments have not progressed much either: I feel that the Smalltalk environments from the late 1970's are the most pleasant, cleanest, fastest, and smoothest PE's I have ever used. While CodeWarrior is reasonably good for C++ development, I haven't seen anything that compares favorably to the Smalltalk systems I used almost 20 years ago. Even the Smalltalk systems of today aren't as clean, easy to use, or well-designed as the original Smalltalks, in my opinion.

We are not even **close** to the ultimate desktop computing/ information/ communication device. There is **much** more work to be done on system architectures and user interfaces. In particular, user interface design must be driven by deep architectural issues and not just new graphical appearances; interfaces are structure, not image. Neither Copland nor Win95 (or NT for that matter) are the last word on operating systems. Unfortunately, the market forces are slowing the development of the next revolution. Still, I think you can count on Apple being the company bringing these improvements to the next generation systems.

I'm sure that some of the things that I remember as having originated at Apple were independently developed elsewhere. But the Mac brought them to the world.

A short bio of Bruce Horn

At Xerox I was a student in the Learning Research Group (1973-1981), where Smalltalk was developed. While there I worked on various projects including the NoteTaker, a portable Smalltalk machine, and wrote the initial Dorado Smalltalk microcode for Smalltalk-76. At the Central Institute for Industrial Research in Oslo, Norway (1980) I ported Smalltalk-78 to an 8086 machine, the Mycron-2000.

At Apple (1981-1984) my contributions included the design and implementation of the Resource Manager, the Dialog Manager and the Finder (with implementation help from Steve Capps). I also was responsible for the type framework for documents, applications, and clipboard data, and a variety of system-level design decisions. I consulted on a variety of projects in the late 80's at Apple as well.

Since then I've received a PhD in CS from Carnegie-Mellon University (1993) and have worked as a computer science consultant, working on projects with Apple and other companies.

This essay is from Jef Raskin.

In the interest of eliciting more historical facts, here are a few additional recollections. A lot of people asked to see my thesis, but that's rather large and much is irrelevant to this discussion, so I've excerpted the pertinent parts of it here.

Bruce Horn pointed out that Xerox had the pull-down menus that I attributed to Atkinson (a case of independent reinvention of which I was unaware) though click-and-drag for selection and moving were (as far as he and I know) invented at Apple. Bruce and others have expressed the idea that the Canon Cat was much as I would have wanted the Mac to be. However, the Canon Cat as marketed by Canon was only a dim echo of what my colleagues and I (at Information Appliance Inc.) worked on. For example, most people only know it as a closed-architecture (which it wasn't) secretarial workstation.

CANON CAT

This is because Canon did not want to reveal that it was actually a 68000-based bit-mapped product with a nice set of graphics tools in ROM--which tools they never used. This was partly because they decided to bundle it with a daisy-wheel printer (!) that could not do graphics and partly because it was brought out by their electric typewriter division and not their computer division. Most observers at the time thought the marketing had been botched, and I am not going to disagree. Designed to allow easy integration of third-party software, Canon never pursued this essential path, and the few third-party vendors we had begun to sign up never had time to complete their work before Canon bailed out. The Cat had a connector and software hooks for a pointing device, but Canon never provided a mouse or mouse-equivalent for it.

In many ways it was, for 1987, far ahead. It automatically resumed from where you had left off working, even if you turned off the machine in the interim; it had a screen-saver; instant on with any keystroke (and you didn't even lose the keystroke); it would have qualified for a "green" sticker had they been invented then, it had true document-centric operation with a level of integration beyond any of today's suites, OpenDoc, or OLE; an ease of use that has yet to be duplicated; the boot time was about seven seconds but seemed instantaneous due to a psychological trick; it featured full-text retrieval of anything, no matter in what application; had built-in communications including an auto-answer, auto-dial modem; the communications were available and integrated with all the application areas; and so on.

Even the normally sage Esther Dyson didn't understand the product's openness, at one point she wrote that it was unacceptable since it didn't do footnotes. But neither did the Mac, until appropriate third-party software came along. But everybody liked the Cat interface.

With regard to my thesis, its formal title was, "A Hardware-Independent Computer Drawing System Using List-Structured Modeling: The Quick-Draw Graphics System" Pennsylvania State University, 1967. All the material in quotes in the next few paragraphs is from the thesis.

Some things I probably should put into their chronological context, otherwise they may seem strange after thirty years of bit-loss. First, I did not have access to an interactive graphics terminal, which I thought would "excite images of a new era in man-machine communications as the more visionary proponents of the interactive console rightly put forth...". Surprisingly, the utility of interactivity was not apparent to all computer scientists at the time; A section of my thesis (6.23) on Interactive Graphics was snuck in nonetheless. For example, when the system had to ask the user something I proposed that small menus could appear right on the display and that the user could "detect on" (now we would say, "click on" as we don't use light pens) the appropriate element in the menu. Now we would call this a dialog box.

I was "providing a common programming system for diverse output media" based on their shared basic abilities, at that time the ability to create a vector. Bit-mapped systems were not available at Penn State. At a time when the existing graphics packages at Penn State drew only "charts, graphs, and tables," I spoke in terms of creating a system for "Architects, electronics engineers, musicians, computer scientists, artists, meteorologists, linguists, chemists, and indeed the entire academic and professional community."

The real need, I wrote, was to "have the ability to define arbitrary symbols and manipulate them into complex pictures. Such symbols could be representations of furniture and fixtures in floor plans, resistors, transistors, and the like in schematic diagrams, notes and clef signs in music, the individual shapes in flow charts, symbols for atoms and molecular structures, sentence structure diagrams, and so on without limit." I saw using images hierarchically, "The fixtures are arranged into rooms, the rooms, now treated as units are arranged into buildings, and the buildings, as units, become developments, urban centers, and cities." The Quick Draw Graphics System (QDGS) provided all affine transformations of objects, and a few others (such as perspective) as well.

For portability it was written in a higher-level language. In this case FORTRAN was chosen, since it was the one scientific language almost universally available at any computer center in the U.S. at the time. True to what I would later do in the industry, I provided a plain-English "QDGS Primer" to get people started, and included it as an appendix to my thesis. Another section of the thesis discussed three-dimensional representation, as the QDGS could do perspective drawings; we made a few short films such as one showing a cube with writing on its faces rotating in space. This is trivial now, of course, but few people were doing 3-D computer-generated animation prior to 1970.

A lot of the thesis was standard computer science / math stuff, with matrix calculations and formal grammars. E.g. "As a grammar... it is context-free, and since it is self-embedding, it is clearly not regular." But I will skip all that formal stuff.

More important from today's perspective is the observation that "with character generators one has a limited choice of lettering sizes... There is one orientation, horizontal, and no ability to introduce new characters." Over the objection that I was sacrificing efficiency I decided that "Within the QDGS no provision is made for the use of character generators, although special programs could be devised. If such provision were made what would be lost, aside from hardware independence, is the ability to have annotations in various sizes, styles, and proportions, at any angle and position. What also would be lost is the ability to treat characters with a full range of transformations available to other geometric shapes, to create arbitrary characters as the need arises, to make annotations part of a figure and thus moved about with the picture it annotates, and the ability to squeeze, justify, fit, and creatively use characters as picture elements in every way."

I went through this same argument again a decade later with Woz when he was designing the Apple II. I argued that he should eliminate the character generator and do all character generation graphically, but Woz didn't think that would work. Jobs didn't understand what was so important about making computers graphics-based. I finally got the hardware architecture I wanted by making it a fundamental principle behind the Mac.

Back to my thesis: to insure hardware independence I considered the effect of different raster sizes and how to compensate in a graphics system for different resolutions "such that if the entire picture is scaled its appearance is the same no matter which device it is plotted or drawn on..." This all seems pretty modern but other things were primitive, "A small innovation... but a nicety, is that labels can be automatically centered." Hard to believe that such an idea could ever have been an "innovation." Many people were surprised to find the graphic arts and typographical term "fonts" in a computer science thesis. Computer scientists were not supposed to be interested in such things, except as an eccentric hobby. Most of my colleagues had never heard of a font. Now, everybody who uses computers is aware of fonts; there's the very word on the menu above this note I am writing.

The most heretical statement I made (my advisor thought it questionable) was that my work was based on a "design and implementation philosophy which demanded generality and human usability over execution speed and efficiency." This at a time when the main aim of computer science courses was to teach you to make programs run fast and use as little memory as possible. Come to think of it, maybe we should bring back some of those courses: nowadays major companies can't seem to write a word processor in fewer than 8 megabytes.

When I put human usability as a major goal, I was off on my own, and did not find like-minded computer scientists until I ran into them at PARC about six years later. In fact, the very term "human usability" didn't enter the computer science lexicon with any regularity until later. The IBM Usability Lab opened in the early 1970's and at first was mainly concerned with ergonomics. Only a handful of people, such as Sutherland, Weinberg, Gilb, and Englebart, seemed interested in the topic and I didn't learn of the work of the last three until after I had done my thesis. (Weinberg's ground-breaking "The Psychology of Computer Programming" was published in 1971." The quantitative work of Card, Moran, and Newell became widely known with the publication of important book, "The Psychology of Human-Computer Interaction". It appeared in 1983.)

My thesis led up to a final illustration: the opening measures from Beethoven's "Variations on God Save the King" for piano. I had created the music font (including an elegant slur generator) and the software for using it as a side project. I also designed and built a digitizer for putting existing scores into the computer, when you pressed a button to indicate that you were pointing where you wanted it noisily punched out Hollerith cards on keypunch machines I had modified. I am not sure that anybody had ever before used general purpose graphics devices to notate music. The notation was not crude, it looked enough like published music so that most people could not detect that it was computer-generated. The story of this work and a photo of the digitizer I built has been published.

As I said in my history of the Mac Project (the one currently being serialized in CHAC), the Mac was by no means the work of one person, but the combined efforts of thousands in hundreds of companies large and small. It was not, as many accounts anachronistically relate, stolen from PARC by Steve Jobs after he saw the Alto running SmallTalk on a visit. For one thing the usual account (as in Levy's book, "Insanely Great" and others) denigrates the original and creative work done by all the Apple employees that put their hearts into the Mac. Most of the histories of the Mac were written without their authors interviewing the original team (Brian Howard, who contributed so much, is always missed), and the history of the Mac that Apple's own P/R department dispensed was based on Jobs's version. Many didn't speak with me: without knowing that I had worked out many of the key usability ideas when Jobs was still in

The creation of the Mac according to Jef Raskin

Friday, August 7, 1998

grade school and before there was a Xerox PARC to learn from, it is perhaps understandable that people would find it necessary to invent a history that derives the Mac's genesis from the nearest similar work. The honest intellectual debt the Mac owes to the work at PARC was not a case of highway robbery.

©1996 by Jef Raskin. Used by permission.



[Home](#)



[StarWay](#)



[Comments](#)

Last Update -- April 17, 1997