

PREPARING A PRESENTATION



MARTIN SMITH¹

You are developing a Human Factors curriculum and you have dictatorial control. What will you do? I myself might count on the CHI Curriculum Committee to produce something soon enough to help structure the course content, and focus my attention elsewhere. For example, I would require that each term, students use a different computer system and a different set of applications to write programs and papers, run experiments, analyze data, record notes, etc. At the end of each term they are to deliver a written account of their experiences with the system, free to gather observations (or data) from their classmates. One graduation requirement would be a paper synthesizing these experiences.

For some people, learning a new application is an exciting challenge, an adventure (or ADVENTURE). Each puzzle along the way is a riddle to be solved. Such people I do not encourage to become system designers -- how could they resist unconsciously including clever twists to challenge those trying to use their programs to get work done? It would be like designing a crossword puzzle with half the answers filled in. For the rest of us, learning an application is an annoying impediment to getting on with our work, undertaken reluctantly if at all. Each puzzle is an irritant, chafing like grains of sand in a running shoe. The point of my curriculum requirement is to teach students to react to such irritants not as would an automobile engine fed a few grains of sand -- by quickly heating up and throwing a rod -- but instead as an oyster might -- by

concentrating quietly and transforming them into something of value.

Each time I've changed jobs I've had to learn new systems, editors, and interfaces. Looking back, I see more thrown rods than pearls of wisdom, and insufficient documentation of the experience. This account of my most recent initiation is an effort to make amends.

The task: preparing overheads for presentation. I have just changed jobs and don't have my familiar tools. My old friend and new colleague, whom I'll call Sean, suggests that I use Microsoft PowerPoint, a Macintosh application designed to create slides, overheads, and hardcopy support for presentations. He hadn't used it, but had heard good things. I was delighted that such a product exists, having feared that I would have to learn a new word processor to get the job done. I figured my simple overheads -- text, with no graphics beyond bullets -- which might take a couple hours to write out by hand, would emerge in half that time from a program specialized for this task. Macintoshes are in heavy demand at my new workplace, but I had an entire day free when I finally sat down at a Mac II and was handed the PowerPoint User's Guide.

Which is a 360+ glossy-paged book. Professionally edited, illustrated, typeset, and hardbound. It suddenly dawned on me that one day might not be enough to make 15 slides.

Admittedly it had been 3 years since I'd used a Mac, and even then not extensively (MacPaint, MacDraw,

¹ Martin Smith is the pseudonym of a scarred veteran of the interface wars, who at times like the above feels more like a battleground than a trooper.

LodeRunner, and Airborne), but I wasn't too worried.² I figured I knew the basics -- if clicking doesn't work, try double-clicking. Unfortunately, no number of clicks seemed to open PowerPoint when I located it in a folder. I wandered the halls and found a Mac wizard, who revealed that I couldn't open it because it was already open -- my first encounter with MultiFinder.

After a few hours I had made enough progress to realize that PowerPoint, once mastered, would be powerful indeed. Unfortunately, the two local PowerPoint experts were not around and several things weren't coming. For example, the title on each slide appeared in what seemed to be the font designed to be read by optical scanners at banks, and despite considerable effort, I was unable to change it.³ Other text appeared in human-readable fonts, but not the title -- which one can't delete from PowerPoint slides. (When I tried to consult on-line help I was informed it was not installed on my machine.)

What to try next? I was working on a Sun, a familiar machine but bereft of the Interleaf desktop publishing software I had used at my last job -- for all its imperfections one of the few pieces of software I have developed true affection for (apart from some games). My Sun has vi, the editor I've used most often over the years. The most interesting characteristic of vi might be how quickly one forgets how to use it when away from it for awhile. In recent years I've used it only to edit ASCII files, principally email messages. (When I typed "v" to summon the email editor on my new system, I was dropped into EMACS, a terrifying experience. I had last used EMACS 10 years ago and couldn't even remember what the "meta" key corresponded to on the Sun keyboard, much less what the command to escape EMACS might be. No one was around to help at the time and soon I was pressing control key sequences at random, eventually reaching a mode where CONTROL-Z ejected me from the flaming wreckage of my message, happy just to get out of there alive.) Anyway, vi without a markup language (or even with one) would be a last resort, short of a typewriter, for producing overheads. Time for another visit to Sean.

"Could you show me what YOU use to do overheads," I asked, "and perhaps make a copy of some that you've done so I can use them as templates?" Templates are a major underdeveloped R&D topic. Perhaps researchers fear the time and relative drudgery of creating template sets of realistic size, or do not recognize that very interesting issues may arise from slowly varying (i.e., monotonous) patterns. Commercial development, at least in my experience, has been slow, perhaps due to the computer memory requirements that template sets impose, but where I have encountered templates, as in the Interleaf system, I've made use of them, appreciated them, and wanted more.

"I use TEDIT on this Xerox LISP machine," Sean replied, and jumped into a rapid-fire half-hour introduction. Sean was like a motorized "training wheels" program: "There are a lot of menus, but here are the only two you REALLY need..." Sometimes these training wheels took me on unnecessary diversions, but this was more than compensated for by the revelation of the occasional invisible object, "...and to get a bullet, type small 'b'" followed by the Expand key..." and the advice that an application manual or help system can never provide, such as how to print files to the nearest printer and where to find transparencies for the photocopier. And, of course, the provision of templates, overheads he had made himself, to work from.

Someone somewhere should grab the last Xerox LISP machine before it lumbers off to the great silicon recycler and install it in a computer museum. It is our duckbilled platypus, a strange mix of the wonderful and wacky, a gold-plated hack. At any rate, the machine I sat down to use had the famous trail-blazing graphical interface -- but when I misspelled a file name, it searched for a very long time, then dropped me into a LISP debugger that required a surprising amount of thoughtfulness to climb out of. When I would freeze in mid-operation for a few seconds, a context-sensitive help message popped up -- an ingenious idea. But I most frequently encountered it when studying a menu to decide which option (e.g. font size) I wanted, at which time the message "will select this item when you release the button" appeared (a message I didn't need to see more than once or twice), completely obscuring the menu I was studying, making it impossible for me to complete my task. (The same message appeared whether the button was already depressed or not).

I have little positive to say about TEDIT, but it got the job done. One good feature is dedicated, labeled function keys for editing operations. (Again, a hit-and-miss implementation: the "Larger" key used to increase font size worked nicely if the next possible larger font was installed on the system, but if not, as was often the case, it searched for a full 45 seconds before returning an error message.) Labeled function keys are such a wonderful thing for the novice that one half regrets the success of the multipurpose PC. I have long awaited the arrival of programmable LCD displays on function keys, or a programmable LCD strip behind a row of them, and I still think there is a fortune there to be had. Another good thing about TEDIT was that it seemed to be very difficult to damage my text by random button-pressing (in contrast, say, to EMACS or even vi, where as we have seen, a few accidental keystrokes may cause the terminal to burst into flames). This was actually more important than it might have been, because perhaps due to flaky hardware or software the system I used had a

² My only uneasiness about the Mac is that after years on Unix, DOS, etc., it feels like working on a tightrope without a net to be unable to drop down to a command interface to bang through some files, however safe one may actually be.

³ I later discovered that what I took to be an odd font was in fact a normal font distorted by the peculiar fact that some large fonts are displayed with what seems lower resolution than small fonts.

tendency to ignore an action the first time I executed it but perhaps carry it out the next, or do completely different things following what appeared to be identical commands. This kind of responsiveness is the norm in dealing with humans but unnerving when engaged in by a computer, and often leads to confused, random button-clicking on my part, remarkably similar to the schizophrenic behavior reported of mammals in double-bind experiments.

I finished at 9 pm, about 12 hours after starting. WYS rarely being WYG from the laser printer -- different wordwrap cut points -- had cost me some time and paper. The photocopier worked fine. Excluding a few hours for lunch and other business, the 15 bullet-and-text slides

averaged about 40 minutes apiece. It was inefficient and frustrating -- and I can't even get class credit for writing about it.

ADVENTURE may be a trademark of someone. PowerPoint is a trademark of Microsoft Corporation. Macintosh is a trademark of Apple Computer, Inc. MacPaint is a trademark of Claris Corp. MacDraw is too. LodeRunner is a trademark of Broderbund, I think. Airborne is or was a trademark of Silicon Beach Software. TEDIT may be a trademark of Xerox. vi may be a trademark of someone. I don't think Richard Stallman would let EMACS be a trademark of anyone.
