

Chapter 8

An Advanced Book for Beginners

How *Computer Lib/Dream Machines* Shaped Our Perspective on Cybercrud, Interactivity, Complex Texts and Computer Creativity

Dick Heiser

8.1 Introduction

Computer Lib/Dream Machines [1] arrived in 1974, exactly in time for the personal computer revolution. It was privately printed and published by Hugo's Book Service. He kept the inventory in his garage.

At The Computer Store, people wanted to find a way to get up to speed quickly about personal computing, but there hadn't been time for mass media to develop a perspective. Enter *Computer Lib/Dream Machines*. We enjoyed recommending it to everybody. The book's enormous size and two-sided format let people know reading it was going to be an interesting adventure. Over our first several years, we sold hundreds of copies. When Microsoft re-issued the book years later, it's clear they tried to civilize it, but everything they tried seemed to backfire. The giant pages resisted being squeezed into a smaller format, and the radical changes Ted Nelson predicted in 1974 were already starting to come about.

Computer Lib/Dream Machines is one of the best examples of a technically advanced book delivering a powerful vision of personal empowerment and enlightenment. It set the tone for the personal computer revolution!

8.2 Cybercrud

We forget how bad the old times were. People were told to do things a certain way "because that's the way the computer requires." Ted was the loudest voice calling out this mistake. He called this thinking *cybercrud*, and *Computer Lib/Dream*

D. Heiser (✉)
Los Angeles, CA 90045, USA
e-mail: dick.heiser@gmail.com

Machines was about empowering users to demand computer systems that accommodate humans rather than the other way around.

Ted pointed out that video games don't have error messages; games just do something reasonable and proceed. The digital dashboard in your car does the same. We have also awakened to the unnecessary annoyance of too many dialog boxes: "Are you sure you want to quit?"

For example, a programmer might phrase a telephone call message: "Illegal number; call aborted!" The telephone company did a better job: "We're sorry, your call cannot be completed as dialed; please check the number and dial again."

We've made such substantial progress in this sphere that we don't have to think about it much anymore. We can look to the Macintosh, iPhone, game console, auto dashboard and Siri for good examples of how to interact with a machine. On the Internet, Amazon, Google, Facebook and Bank of America operate outstanding websites that are flexible, capable and a pleasure to use. This is huge. There are still bad websites and bad software, some of it spectacularly bad, but the example of the good ones will drive out the bad ones.

8.3 Interactivity

David Albrecht of People's Computer Company (PCC: what a radical name!) discovered and promoted *Computer Lib*. People's Computer Company operated a timesharing BASIC computer lab in Menlo Park, and published a newsletter on interactive computing. The newsletter told me how to get the book. PCC also published a big book of computer games in BASIC, called *What to Do After You Hit Return*. One guessing game was called "Hunt the Wumpus." It was lucky for Ted that Bob Albrecht knew about *Computer Lib*, because Hugo's Book Service had few contacts among computer enthusiasts.

Ted also chose only interactive interpreted languages to explain programming: TRAC, APL and BASIC. This is an amazing and powerful way to give a perspective about programming. Ted also discussed the importance of simulation and graphics, of course.

8.4 The Hands-On Imperative

One of the biggest changes has been in the way we see computers. In the 1960s computers stood on raised flooring behind locked doors and you needed an account number, meaning a serious business purpose, to participate. In fact, if you weren't a big organization dealing with lots of money, you'd be told to forget using computers anyway. Computers were for math rather than for literature or connecting with others or discovering ideas. Ted's book awakened people to a bigger vision.

The personal computer revolution was a very emphatic experience for those of us who considered ourselves activists; we knew hands-on computing was a big deal and we felt the pushback from the computer industry establishment. We called it a revolution and it really was one.

People's Computer Company was way ahead of its time. It featured the "hands-on imperative." Starting in 1975, user groups in Los Angeles and the Bay Area educated people and helped them find answers and resources. Byte Magazine was started; conventions, swap meets and newsletters spread the word. Everybody can participate. We have layers of powerful, responsive, computing resources from handheld to the cloud.

8.5 Complex Texts

Rectangular tables of data are not the only way to organize information in a computer. Lyall Morril developed *Whatsit?* a freeform information organizer that used triples to record relationships between entities. That was a little step toward loosening up people's thinking in the direction Ted was and is advocating.

I am especially grateful to Ted for introducing me to Douglas Engelbart, another amazing visionary, the man who gave "the mother of all demos." Engelbart showed creative ways of organizing work and ideas, and of collaborating online.

An attorney customer of ours created a program to organize legal arguments. His program let a user connect evidence to arguments and arguments to evidence. Primitive personal computer languages made it difficult to store text strings longer than 256 characters, but even with those limitations, the program worked well.

8.6 "Everything Is Deeply Intertwined"

The quotation that serves as the heading for this section appeared on page D2 in the *Dream Machines* half of Ted's book, *Computer Lib/Dream Machines* [1]. Ted was key to the development of hypertext, but he's disappointed that we didn't make links that work in both directions. Even one-way links can be amazing. I had a transformative experience with a Hypercard program called *The Dungeon of Class Gifts*. It trained lawyers about the rules for group inheritance. It's very light-hearted and breezy. After each page of explanation, you're asked a question with two possible answers. If you choose the right answer, you advance to the next screen. Choose the wrong one, and you're sent to a screen picturing an explosion. The screen says "a mind is a terrible thing to waste" and then the computer sends you back to the beginning of the lesson. Gradually you progress farther into the sequence. By the time you're nearly done, you have answered the first questions many times, and you are energized; you're in a hurry to finish the lesson. It's basically a game of Parcheesi!

The game forces you to rehearse the facts enough times to learn them. It's a delightful way to be seduced into memorizing something!

8.7 Xanadu

It was fun to hang out with the Xanadoodlers, especially in the 1970s and 1980s. Xanadu is a daring design that presented awesome challenges: how to link to evolving documents, how to track changes to a document, how to manipulate linkages, how to organize archival storage, how to name the target of a link, how to track micro-copyright royalties, how to organize the physical storage of a universe of discourse and how to scale storage and processing around the world.

Many people are still skeptical of the need for bi-directional links. I am one who suspects links might only occasionally need to be bi-directional, or that a pair of one-way links could simulate a bi-directional link.

Claude Shannon's popular demonstration of his computer-controlled maze-navigating mouse was essential to the success of his project. Shannon's demonstration appeared as a segment in the television show, *Time Machine: Robots* [2]. Shannon went to a lot of trouble to prepare a tabletop maze and to eliminate any arm or cord connecting the mouse to the computer. This greatly enhanced the impact of his presentation, and of his theory. It helped Shannon gain traction with his audience.

Similarly, Xanadu needed terrific examples of information organized with Xanadu using bi-directional links, examples of change tracking, annotation and link navigation. This would have helped less-abstract thinkers like me understand the system better. Showing how to write a response to a 100-page Request For Proposal would give Xanadu a chance to show off.

Many problems still perplex even fans of Xanadu. I can't imagine how to organize the millions of links that might want to connect to popular texts like the Bible or the U.S. Constitution.

It was a constant struggle to try to implement Xanadu on the computers available at the time. Hardware was way too expensive, too small and too slow. Nowadays, we could prototype the system with JPEG screenshots. Google's massive datacenter technology would be just the ticket for running Xanadu.

8.8 Computer Creativity

We had a poster in our computer store quoting Ted about the creative possibilities inherent in interactive computing. I re-read *Computer Lib* looking for this quote, and found many surprises, including the *Computer Lib* pledge (Fig. 8.1). I also found where I learned many of my basic perspectives about computers.

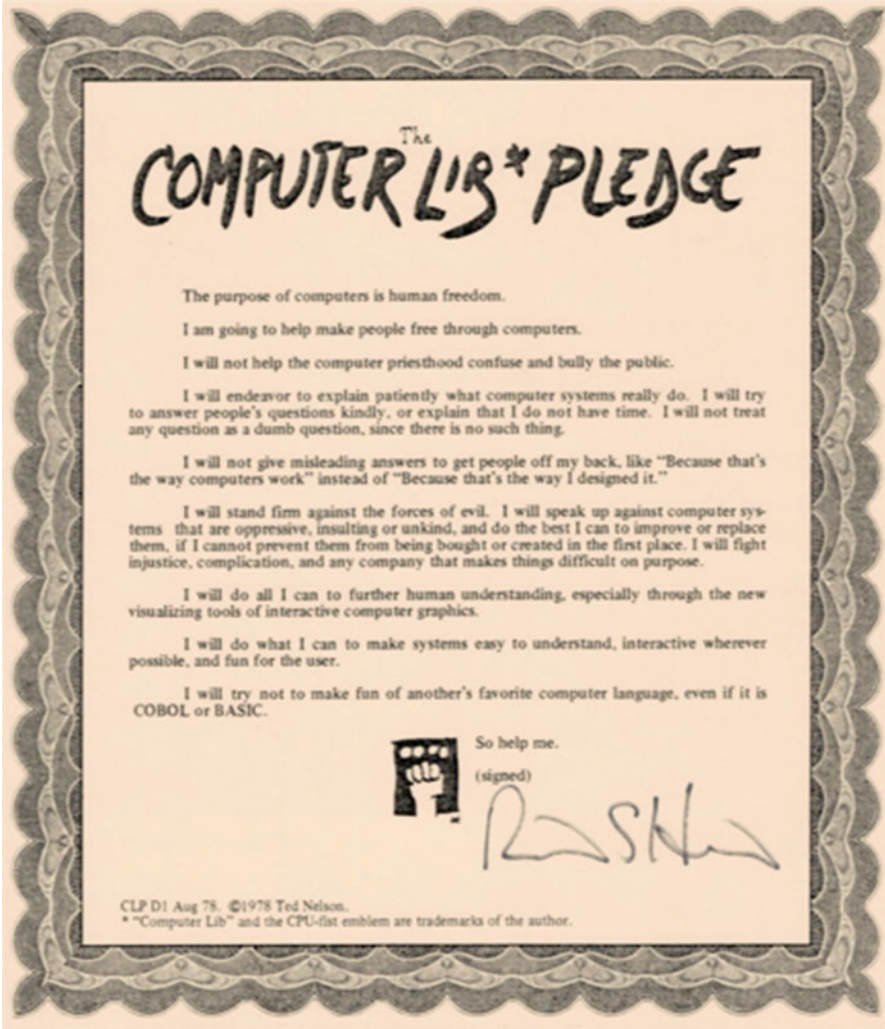


Fig. 8.1 The *Computer Lib* pledge

Computer Lib was just the right advice for someone shopping for a personal computer in 1975! It's an advanced book for beginners!

Open Access This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Nelson TH (1974) Computer lib: you can and must understand computers now/dream Machines. Hugo's Book Service, Chicago
2. Time Machine: Robots. *Robots*. History Channel. Aug.–Sept. 2000. Television. See segment on Claude, Shannon. 1952. *Theseus Maze-Solving Mouse*. (begins at 9:16 in the video). <http://youtu.be/KmURvu4x0Do>