

# Microware Review

Len Proctor

---

## Apple's HyperCard

Reviewed by Earl R. Misanchuk

---

Before I actually tried out HyperCard, I had read a number of reviews of it in the popular press. What intrigued me about those reviews were the waffle-words that were used to describe what kind of software HyperCard was. My expectation, based on those reviews, was that HyperCard was sort of a database program, but not really; much like hypertext, but not exactly; kind of a programming language, but not really a programming language because even non-programmers could understand it. And oh, yes, it contained all of MacPaint, plus a number of features MacPaint didn't have.

Now that I've had an opportunity to actually work with HyperCard, I can unequivocally state that HyperCard is sort of a database program, but not really; much like hypertext, but not exactly; kind of a programming language, but not really a programming language because even non-programmers can understand it. And oh, yes, it contains all of MacPaint, plus a number of features MacPaint doesn't have.

Rather than attempt to force-fit HyperCard into one of the extant categories of software types, let me try to describe one or two uses to which HyperCard can be put. Then perhaps you will agree with me that HyperCard is *sui generis*. (Whether it will remain the only exemplar of its class or whether competing products will develop to challenge it is a topic of much debate on the computer networks today.)

The name HyperCard implies the metaphor on which the program is based: a stack of cards, each containing information. They are not ordinary cards, however—they are definitely “souped up” enough to deserve the prefix “hype? Cards can contain text, graphic elements, buttons (“live” spots on the screen which, when clicked upon, cause something to happen, e.g., movement to another card or stack), and simple sound. Users can move from one card to the next in either direction through the stack, or to another card elsewhere in the stack. Furthermore, the user can blithely move to a card in another stack, then back to the first stack, with absolute transparency. The movement can be done under the control of the user or of the author of the stack.

Let me describe a few stacks that come with the program, in hopes of giving you a

---

Len Proctor is an Associate Professor of Educational Communications at the University of Saskatchewan, Saskatoon, Saskatchewan.

---

fuller picture of what HyperCard is and can do. There is a basic rolodex emulator, which can be used to store addresses and phone numbers (a standard database-type application). It can also be used, in conjunction with a modem, to actually dial the phone numbers for you. To use the **rolodex**, users simply add or modify text to the on-screen representations of rolodex cards, using standard Macintosh editing techniques. Of course, cards can be added, deleted, copied, etc. MacPaint-like graphics can be used to add visual components to cards, as well.

There is a calendar which, when you click the mouse on a particular date, becomes a weekly calendar book for the week containing that date. This ability to bring to the fore some related information by clicking on a certain spot (called a button) is strongly reminiscent of hypertext programs (e.g., OWL's Guide). Also linked to the weekly calendar and yearly calendar is a To Do list, Click on a certain date and the To Do list for that date appears. This ability to link the three separate stacks together, functionally, is reminiscent of a relational database program.

There are a couple of other idea-generating stacks that illustrate what kinds of information can readily be stored in HyperCard form, using quotations and clip art as the subject matter. Some of the 19 assorted stacks that come with the programs illustrate a few of the other things HyperCard can do: It can store graphic images as readily as text; it can sort cards according to criteria you specify; it can flip rapidly through all the cards in a stack; it can search through a stack for a user-specified keyword(s) with incredible speed, without any need for limiting the search to specific fields. Indeed, one of the reasons that HyperCard may well continue to be unique for some time is the fact that it contains new, copyrighted algorithms for extra-rapid searching. How does a three-keyword, field-free search through 50,000 cards, in under two seconds sound? (I haven't done it myself, but a usually reliable source quoted those figures to me.)

One of the stacks, the Help stack, is a sight to behold. It is unquestionably the finest example of internal documentation these tired eyes have ever seen. The obvious care given to its instructional design is typical of Macintosh program manuals, but not at all typical of help files, on the Mac or (especially) in the MS-DOS or mainframe worlds. Profusely illustrated, well explained, and demonstrating many of HyperCard's powerful features itself, it can be a more than adequate introduction to the program by itself, if for some reason you can't locate your manual.

The power of HyperCard begins to become evident when the user progresses beyond using existing stacks and begins customizing them for specific uses. For example, the first thing that came to mind when I explored the sample stacks was HyperCard's potential for doing literature searches (in the sense of finding common elements within summary notes made on various articles and books). As I experimented with HyperCard, my eyes drifted upward to the spot on my shelf where my 15-year-old collection of key-sort cards languished. Given HyperCard's rapid search capabilities and freedom from restrictive fields, I almost wanted to go back to being a graduate student again. (I suspect I risk the wrath of today's graduate students when I say "It would be so easy now!")

But the real power of HyperCard shows itself only when the user progresses to the level of what the HyperCard manual (which itself is a typically Macintosh, high-quality

instructional device) calls authoring--creating new HyperCard stacks from scratch. The manual covers the basic procedures of stack creation adequately, but only hints at what can be done if the author is willing to venture into the realm of HyperTalk, the programming language that forms an integral part of HyperCard. What can HyperTalk do? Too many things to go into detail on in this review, but all with a very English-like language. Does the following sound like programming?

```
on mouseUp
    visual effect scroll left very fast
    go to card "RobertFrost" of stack "Poets"
end mouseUp
```

Or does this sound like programming?

```
on mouseUp
    if the first character of the message box is in "0123456789("
        then dial the message box
    else
        Ask "Dial what number?"
        dial it
    end if
end mouseUp
```

Well, maybe just a bit. But notice how easy it is to understand? HyperTalk will happily execute those instructions.

Simple animation is possible. So is sound (even simple music). Question asking and answer checking. Card and stack protection, with or without passwords. So are reading files, launching applications, and a great deal more. Easily, without great gobs of time spent learning how to program to get it.

ASIDE: If you are serious about wanting to get involved in authoring, you should acquire Danny' Goodman's hefty (720 pages) paperback, *The Complete HyperCard Handbook* (Bantam). The first half of Goodman's book provides a more in-depth treatment of HyperCard's features than does the actual HyperCard manual, along with many hints and shortcuts. The second half is devoted entirely to HyperTalk. The book is very well written, in language understandable to a non-programmer, even the part on programming (obviously, some experience in programming certainly wouldn't hurt, however).

Does this all sound too good to be true? Here's the final straw: Apple now gives away HyperCard with every purchase of a Macintosh Plus or Macintosh SE. If you've owned your Mac for a while, you can buy the program from Apple dealers for around \$50 (educational price).

There has to be a catch, right? Well, maybe a small one: HyperCard will only run on a Mac, and then only on a Mac with at least one megabyte of memory - more would be better. In addition, stacks can consume huge amounts of storage space (the Help stacks that come with HyperCard are a whopping 736 kb) so a two-800K-drive

system is an absolute minimum, and a hard disk is highly advisable. Don't even dream of trying to use it with a single-drive Mac. You'll end up with "swapper's elbow" from inserting and removing disks so many times.

HyperCard does have room for improvement, in a number of small ways. It would be nice, for example, to mix italic and bold type with normal type (as it, you can make all the type in a field normal, italic, or bold, but you cannot select certain words to have those characteristics). But that's quibbling. The poor educator's mind boggles at the potential of this program as a learning environment, even in its current incarnation. Imagine a stack with the first card having a map of the world. Click anywhere on France, and a map of that country fills the screen. Click on Paris, and get a street map. Move around the map by clicking arrows for adjacent close-up maps until you locate the Louvre, and click on it; the screen fills with a menu of artists' names. Any name selected is replaced with a pop-up menu of titles. Finally, choosing a single title brings a graphic of the work of art onto the screen.

Couple. HyperCard with an optical or digital disk (which of course, can be done easily), and whole worlds of information-delivery potential open up.

Keep an eye on this program -it's going to have a profound effect on how information is stored, disseminated, and (especially) used. A number of share-ware stacks on a wide variety of topics are now available on computer networks; more will be forthcoming over the next 12-18 months. Several commercial stacks are now available, and we can expect a great many more of these soon.

The author of HyperCard (Bill Atkinson, who created MacPaint) wanted to put programming power into the hands of non-programmers. I think he's succeeded. It's an exciting time to be an educator.