leagues at the College de France in Paris. Combining what he calls supramolecular chemistry and catalysis, Lehn is working on a system of molecular microreactors and artificial cells. These new reactor systems would selectively produce novel and complex chemical products—advanced drugs, plastics, fibers, composite materials, and fuels. In a more refined state of development, these revolutionary reactor systems would be the chemical analog of computer circuitry and systems. A hierarchy of cells that recognize, store, and treat information at the molecular level may be the future's molecular chip. S

SOFTWARE'S SECOND ACT

The next strides in computers will come from novel architectures.

BY ALAN KAY

In a puppet show, representations made from lifeless material are manipulated so adroitly that they seem alive and full of purposeful character. Only part of the presentation is seen by the audience; the rest is managed offstage by the puppeteers, costumers, scene builders, and playwright. If the puppets were letters and the stage set made to look like a computer screen, we might imagine the puppeteers dancing letters across the screen Busby Berkeley fashion to form words, sentences, entire messages. If someone in the audience could tell the puppeteers to move one word off the stage and replace it with another, then we would have a striking analogy to what actually goes on in word processing on a personal computer. The letters that look like marks of ink on the screen are actually costumes worn by the many thousands of nondescript players of this newest form of theater.

The computer is easy to understand if we realize that everything it does is guided by a script. There are no important limitations to the kinds of plays that can be acted, nor to the range of costumes or roles that the actors can assume.

We say that the theater—even puppet theater—simulates realities rather than imitates reality because ideas about things that have never happened in the real world can still be acted out realistically. As in theater, people who interact with computers quickly form myths to explain and predict the action taking place before them—in this case, on the screen. In theater, the audience will identify emotionally with what they see, bringing their own experience to bear on the action, and they will then ignore the fact that the story being related is not real. Similarly, if a computer program is scripted to simulate a real action, the user will try to believe that the simulation is real. For example, there are popular ways to interact with the computer interfaces that allow a user to get rid of an object shown on the screen by placing the object in a trash can. The computer user assumes that later, should he choose to look into that trash can, the object will still be there. If it is, then the designer of what is called the "user interface" has followed a tradition that antedates the Greeks by setting a scene that successfully shapes and bounds the user's myth.

As with the audience of a play, the user's joy, power, and satisfaction will depend on the software designer's skill in creating and consistently maintaining the myth throughout the life of the "play." Magical happenings that don't seem to be possible in the objective world are quite acceptable as long as they obey a consistent logic that permits prediction.

When users of computers can directly manipulate those myths, they can be said to have "leverage." If, for example, guesses work out, if nudges of a computer "mouse" move things, if animated figures can be conversed with, the user has far more power to accomplish things. In this way, the audience can be a co-conspirator with the scriptwriter.

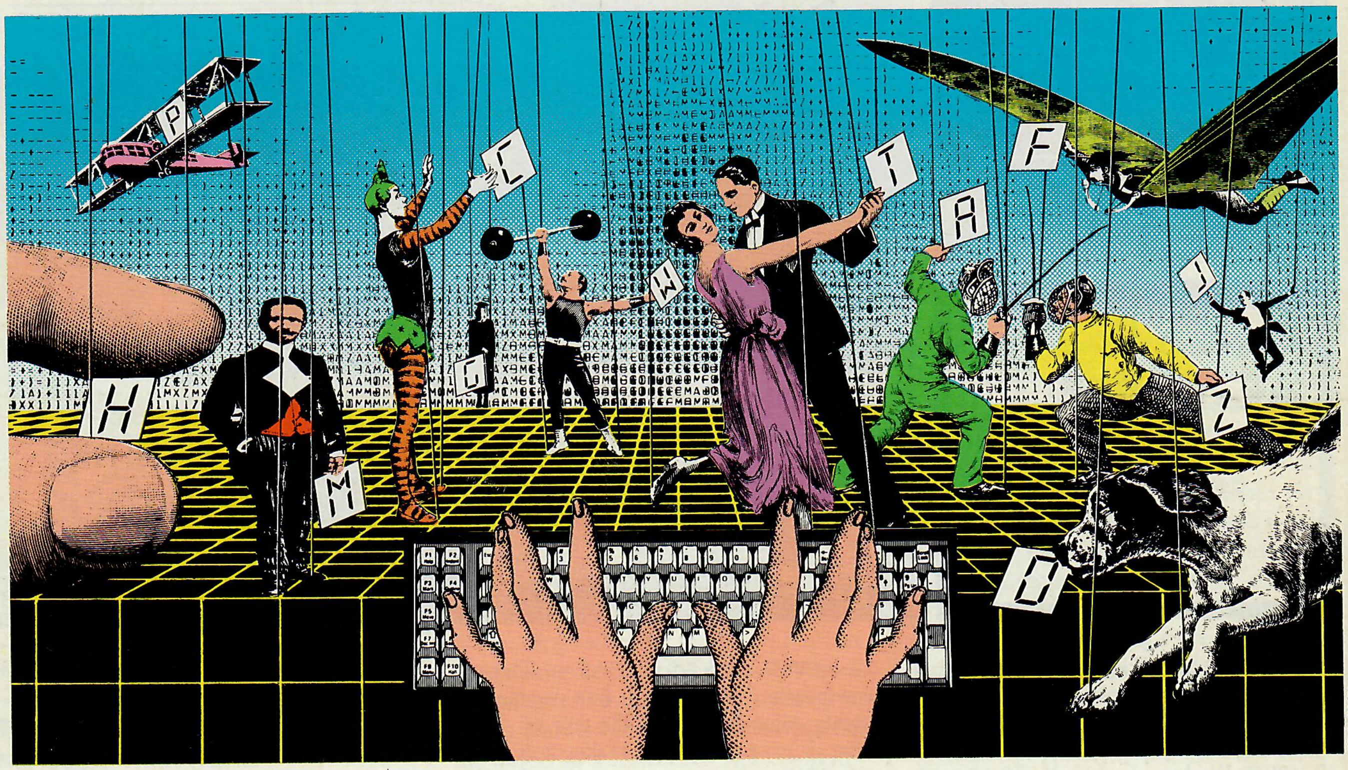
But there is still the question of how much one needs to know or to learn to carry out different functions. Are we after a script no more sophisticated than one of Punch and Judy, or are we after the computer programming equivalent of Shakespearian drama? In fact we want the interactive experience to steadily increase users' sophistication and ability to deal with their information structures—what appears before them on the screen—from ever richer perspectives. Still more of a challenge, the user interface and functionality must keep pace with changes in each user's knowledge.

Finally, trivial content just won't do, regardless of how well its assumptions hold together. An easy-to-use word processor, for example, is useless if it doesn't provide the functions required by its user. For example, the user might want to write in a Hollywood script format, and the word processor just won't do it. This would not be so important if there were ways for the user to tell the word processor what new properties it should have. But today's user runs into trouble because the languages used to write the internal scripts of computers are cryptic beyond decency; they are less capable of expressing human-level ideas than the sorriest pidgin ever coined.

Most of the leverage enjoyed by users to this point has been *mechanical.* Word processors, drawing systems, spread sheets, all extend our reach in the same manner as pencil and paper, carpenter's tools, and telescopes. Tool extension—methods that would allow the user to customize what today is rigidly programmed—will flourish mightily in the next decade. For example, master designers will create the best word processor they can think of and give it to the users as a kit. The script will be written in a language users understand, such as English, and they will be able to add or subtract details to fit the word processor to their precise needs.

Managers of small businesses, who usually can't afford the services of a programmer, will then be able to "sculpt" an information-handling system to fit their needs. They will use special tools to draw the kinds of screens they wish to see and provide rules that stipulate the transactions to be performed. Much of this will be accomplished by indicating changes in those basic kits.

But even the best direct aids to user programming will become cumbersome



JOHN PACK

in the face of the enormous increase of resources and pathways made possible by tomorrow's "information utilities." Like today's energy or telephone utilities, information utilities will offer an immense variety of services, each of great complexity. As a result, in a few years a long-contemplated, entirely different way to interact with computers will have to be worked out.

Today the objects on the screen are representations of information; the user is the direct agent of change in the search for, display, and processing of information. But we will be overwhelmed by such tasks tomorrow and so will need to delegate most of our information gathering and forecasting tasks to a squadron of assistants—something like secretaries who know how to skillfully handle tasks for us. These electronic secretaries won't be so much manipulated as managed by us.

For example, our small business managers will not sculpt so much as direct. They will explain their needs much as they would to a programmer if they had one: by giving examples, by diagramming typical transactions, by indicating differences between their ideas and the kinds of structures the system is set up to initially handle. Systems like these will be capable of some learning. But most of the systems we'll see in the next few years will be constructed like today's expert systems that can find new petroleum deposits on the basis of survey data. Yet, just as bees execute their social rites in primitive ways, these expert systems accomplish their impressive feats in a plodding manner.

Of course, all this implies considerably more computing resources than personal computer users enjoy at present—but "more and faster" are easier next steps in the computer field than "new and better." Recently there has been a noticeable slowdown of pivotal research and breakthroughs due to a combination of rapid commercialization of personal computers, overly rational government spending policies that shun research risk, and perhaps the fact that many of the easy problems are now worked out. Thus, most of the advances in the next five (perhaps 10) years are likely to be based on laboratory work that can be transferred to the marketplace as silicon chips continue to dive in cost while skyrocketing in capability.

These advances will be considerable. Quantitative changes by factors of 10 in almost anything—in speed and size, say, of computer memory—*feel* like qualitative changes. Since the 1960s, the computer industry has seen changes by a factor of 10 million, and we can expect at least two more improvements of this magnitude before 1995. For special computing tasks such as 3D graphics, speech generation and recognition, and music synthesis, we can expect factors not of a hundred but of a thousand or more. Many of the consequences will appear wonderful to a generation raised on special effects. For example, there is every reason to expect that a $500 plug-in card will be available in 1995 that will produce real-time animation of a quality that requires hours per frame on the fastest super-computers of today.

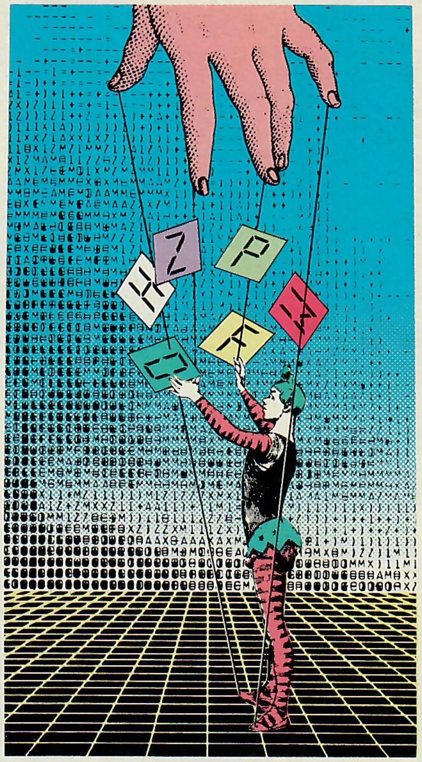
Even though in theatrical terms, most of these advances will be in spectacle rather than in content, I admit that it will be great to have the capability of showing objects in 3D, and a unit that understands idiosyncratic speech, and a computer memory that might store a few bookshelves worth of memory, and high-bandwidth fiber networks that can rapidly carry huge numbers of video, audio, and data channels in and out of the home. Indeed, the wonderful ideas of Vannevar Bush, the computer pioneer who served as head of the Office of Research and Development during World War II, may soon be realized. In 1945 he proposed a memex, a home machine that could store the contents of a 5,000-volume library and allow you to perform creative research with ease. It is still a great dream and still needed.

The most unnoticeable change in the next decade will be the proliferation of computers to the point where they become utterly unremarkable. Like pencil and paper, the telephone, and other technologies that have become a way of life, the computer will be noticed only when absent. Marcian Hoff, the inventor of the microprocessor, once imagined someone in 1905 exclaiming, "You know, the electrical motor is a wonderful thing; every home should have one." This conjures up visions of a big motor in the attic running everything in the house by belts. But it didn't happen that way. Today's typical home has more than 50 electric motors, all invisible. When memex arrives, it will be in the walls of our house and woven into the fabric of our clothing—and our lives.

The most exciting next step for computing in the coming 10 to 15 years will be to discover the DNA of computer mentality and build architectures that give rise to recognizable intelligence and learning. All the great increases predicted for hardware resources won't do any good. We simply don't know how to build these advanced structures.

Certain philosophers who have been (with some justification) critical of the claims of the last 20 years say that it can't be done, that the only way you can make a machine think like a brain is to make a brain. But these philosophers admit that suitably arranged atoms can think. And they then put forth two main claims, one I think true and the other not.

The first, with which I concur, is that much of the character of human intelligent thought is tightly coupled with our innate behavior, the peculiar fashion in which neurons deal with information, and the kind of cultures in which we are brought up. That means it is highly unlikely that current artificial intelligence methods will approach human levels, though mimicry will abound.



JOHN PACK

If computers are thought of as electronic puppet shows, there are no important limitations to the kind of plays that can be enacted on their screens, nor to the range of costumes or roles that the actors can assume.

Their second premise is that true intelligence can only be attained in a machine by detailed simulation at the neurobiochemical level. I believe this to be quite without foundation. One reason is that although Mother Nature has been capable of remarkable things, she is a Rube Goldberg designer of the first rank. This is especially true in biology where much of the complexity and fragility of living mechanisms has to do with how low the energy margins are that carry out reactions and how accidental the efficiency of those reactions is.

When we do organic chemistry in the lab, however, we have two things going for us that nature doesn't: a clear goal and the ability to bring lots of energy to bear on a task. Similarly, in computer science we should really expect to accomplish mental operations with a lot less machinery than nature requires.

It is likely that the strategies for memory, learning, and consciousness can be worked out by using as a model the parsimonious reuse of structures such as neurons and the positively stingy amount of DNA for specifying organisms. Most higher mammals—the rat, the cat, and us, for example—have about the same amount of DNA per cell. Much of the evolution of the human brain was accomplished by a vast replication and folding of the neocortex—compared to the mouse, for example, humans have a factor of 1,000 or more total brain cells, perhaps a million times as many synapses. But the same neurons served for the crayfish and for Bach; the notes are the same but the melody and counterpoint infinitely richer. Our innate wiring for facelike images or language acquisition can't use much more DNA than the circuitry of less able mammals.

Instead, the die is cast for lots of memory, a stable culture, and long learning. When the tapestry is finally turned to reveal its pattern, it will be one we can weave ourselves. ■