

beyond the box

Alan Kay believes a quantum leap in personal computing is long overdue. But it will require a whole new way of thinking.

ALAN KAY IS NOT A FAN OF THE personal computer, though he did as much as anyone to create it. A winner of the Turing Prize, computer scientist Kay was the leader of the group that invented object-oriented programming, the graphical user interface, 3D computer graphics, and ARPANET, the predecessor of the Internet. After helping to create the Alto, the Xerox PARC PC prototype that inspired the Apple Macintosh, he took on the role of chief scientist at Atari Corp. and became a fellow at Apple Computer, Walt Disney Co. and Hewlett-Packard Co.

While most people regard the personal computer as a modern miracle, Kay sees the PC as a chronic under-achiever. To him it's an invention that, like television, has fallen far short of the potential foreseen by its early proponents. Today, at age 66, Kay runs the Viewpoints Research Institute, his

own nonprofit research organization in Glendale, Calif. He is busy with several projects involving education and technology, including the "One Laptop per Child" project overseen by MIT's Nicholas Negroponte, which Kay hopes will one day transform the PC into a machine that not only has changed the way we work, communicate and entertain ourselves, but improve how people—especially children—learn and think.

Kay believes the limitations of the PC are due as much to lack of imagination and curiosity on the part of computer scientists, the unwillingness of users to invest effort into using computers, and the deadening impact of popular culture, as they are to technical constraints. He says the push to make PCs easy to use has also made them less useful; their popularity has stunted their potential. Executive Editor Allan Alter spoke with Kay about the future of the PC. The following is an edited version of their discussion.

CIO INSIGHT: Do you feel PCs and Macs have come close to reaching their potential yet?

KAY: No, I don't think so. Computers are mostly used for static media, basically text, pictures, movies, music and so forth. The Internet is used as a distribution network, so computers are essentially players for this media. This is incredibly useful, but it tends to overwhelm uses that require a much longer learning curve.

When I started in computing in the early sixties, people realized

what personal computing should be about. And on one of the early hits you can watch the demo he gave in 1968 to 3,000 people in San Francisco, showing them what the world of the future would be like.

Engelbart, right from his very first proposal to ARPA [Advanced Research Projects Agency], said that when adults accomplish something that's important, they almost always do it through some sort of group activity. If computing was going to amount to anything, it should be an amplifier of the col-

whether they've ever typed E-N-G-E-L-B-A-R-T into Google—and none of them have. I don't think you could find a physicist who has not gone back and tried to find out what Newton actually did. It's unimaginable. Yet the computing profession acts as if there isn't anything to learn from the past, so most people haven't gone back and referenced what Engelbart thought.

The things that are wrong with the Web today are due to this lack of curiosity in the computing profession. And it's very characteristic of a pop culture. Pop culture lives in the present; it doesn't really live in the future or want to know about great ideas from the past. I'm saying there's a lot of useful knowledge and wisdom out there for anybody who is curious, and who takes the time to do something other than just executing on some current plan. Cicero said, "Who knows only his own generation remains always a child." People who live in the present often wind up exploiting the present to an extent that it starts removing the possibility of having a future.

"The things that are wrong with the Web today are due to this lack of curiosity in the computing profession."

that while the computer could simulate things we understood very well, one of its greatest uses was simulating things that we didn't understand as well as we needed to. This has happened in the sciences; physicists, chemists, biologists and other scientists could not do what they've been doing if they didn't have powerful computer simulations to go beyond what classical mathematics could do. But it's the rare person who quests for knowledge and understanding.

A great thinker in our field is Doug Engelbart, who is mostly remembered for inventing the computer mouse. If you search Google you will find Doug's Web page, where there are 75 essays about

lective intelligence of groups. But Engelbart pointed out that most organizations don't really know what they know, and are poor at transmitting new ideas and new plans in a way that's understandable. Organizations are mostly organized around their current goals. Some organizations have a part that tries to improve the process for attaining current goals. But very few organizations improve the process of figuring out what the goals should be.

Most of the ideas in that sphere, good ideas that would apply to business, were written down 40 years ago by Engelbart. But in the last few years I've been asking computer scientists and programmers

In addition to a mindset that prevents real innovation, is the technology inside the machine itself a problem?

The types of disk drives we have are slightly faster than Moore's Law predicted for silicon. Costs have gone down pretty well. But there are bottlenecks. The architectures have not advanced near to Moore's Law proportions. Possibly, back in the mid-1980s, all the machines that were on the Internet put together were approximately equal to a top-of-the-line desktop or laptop computer today. But if you look at how the Internet has

scaled since then, the operating system and other software inside your machine hasn't scaled well. Why doesn't my machine look like an Internet inside? Because the Internet is a scalable architecture made up of real machines that are made up out of virtual machines. And that concept could actually be used as a basis for an operating environment that essentially doesn't have an operating system.

There isn't a particularly good reason this hasn't happened; it's just that the dominant operating system architectures that we have are all from the sixties. Basically, the people who do operating systems got used to this kind of layered architecture in an operating system, and they tend to keep on feeding it, even though layered systems don't scale very well. This is an example of the invisibility of normality. We're not even aware that we're accepting most things we accept. Any creative person has to try and force their brain to reconsider things that are accepted so widely they seem like laws of the universe. Very often they aren't laws of the universe; they're just conventions.

The Viewpoints Research Institute is supporting several technologies and projects aimed at inventing fundamentally new ways of computing. What are Squeak and Croquet, and are there others?

Squeak, an object-oriented operating system and authoring environment, is actually the Xerox PARC Smalltalk operating system, upgraded to 32-bit graphics with other things added. My research group at Apple did it about ten years ago, because we were afraid that Java wasn't going to be com-



patible from computer to computer. Because we made our own software tools at Xerox PARC, and we had some of the same people who had done these tools, we decided we'd be much safer if we just made our own vehicle. At Viewpoints and Hewlett-Packard, we built an operating system for children, and also did many experiments in user-interface design and built new kinds of object models and other kinds of things called Etoys.

You can think of Croquet as a new way of doing an operating system, or as a layer over TCP/IP that automatically coordinates dynamic objects over the entire Internet in real time. This coordination is done efficiently enough so that people with just their computers, and no other central server, can work in the same virtual shared space in real time. It's mature enough to be supported by the Open Source Foundation and there is a start-up company called Qwaq that is working with Croquet. (I'm not part of the start-up.)

We brought together David Reed—Croquet is a working model of his 1970s thesis—with other talented people, and funded them.

The Viewpoints Research Institute is actually involved in three new projects. One is the \$100 laptop project that Nicholas Negroponte is doing. That is coming along very well. The first 1,000 factory-built machines were built in the last few weeks. The plan is to build 5 million to 8 million laptops this summer, and perhaps as many as 50 million in 2008. We're very involved in that. The other thing is a recently funded NSF project that will take a couple of giant steps, we hope, toward reinventing programming. The plan is to take the entire personal-computing experience from the end user down to the silicon and make a system from scratch that recapitulates everything people are used to—desktop publishing, Internet experiences, etc.—in less than 20,000 lines of code. It would be kind of like a Moore's Law step in software. It's going to be quite difficult to do this work in five years, but it will be exciting.

The third project we're just getting started on and don't have completely funded yet, is to make a new kind of user interface that can actually help people learn things, from very mundane things about how their computer system works to more interesting things like math, science, reading and writing. This project came about because of the \$100 laptop. In order for the \$100 laptop to be successful in the educational realm, it has to take on some mentoring processes itself. This is an old idea that goes all the way back to the sixties. Many people have worked on it. It just has never gotten above threshold.

How would these new systems and computers be different from the kinds of PCs we're familiar with now? I don't want to get hyperbolic about it, but one of the most interesting questions connected with this is: How much learning is a person willing to do to *really* learn how to use a computer? The answer, over the last 25 years of the commercialization of personal computing, is almost none. Nobody really wants to put in any amount of effort. The things that people have been

the newer ideas about what computers can do.

The spreadsheet, for example, with a few changes in it, would be thought of as being a highly parallel simulation engine. If you think of the purpose of the spreadsheet being not only to tabulate what did happen, but to give you an idea of what could happen, you would immediately redesign the spreadsheet and integrate it with graphical displays or visualization in a very different way. You would be

“How much learning is a person willing to do to *really* learn how to use a computer? The answer . . . is almost none.”

willing to learn have tended to be like the media they grew up with, which have really simple user interfaces. (The big exception is video games.) You don't see Doug Engelbart's approach to user interface, which was an incredibly efficient, two-handed interface that required training to learn how to use.

One way of looking at personal computing is to focus on the kinds of things that computers can help people learn. There are a whole bunch of things that can be done if learning, rather than function, matters. If you were to change the approach to the user interface, as we thought we were doing at Xerox PARC, to a more learning-curve-oriented system, then you would be able to accelerate the acceptance of

on the road to a different kind of computer literacy. Another area is trying to keep track of lots of things that are happening simultaneously in time, and building a language that would model an aspect of the real world that's so important to us. There are just dozens of these examples.

Is it really necessary to reinvent the PC if computing is to leap forward? When you go to a mass market and marketing is driving things rather than ideas, customers and the marketers will find a common ground where the marketers and buyers don't have to work too hard. The mundanity of the market sustains itself for a while, but then starts actually hurting

itself because it gets into a position where it cannot imagine what the next stages would be.

What could get people to adapt this new style of computing?

Basically, the reason I work with children and not adults is because adults are famously difficult to change in any significant way. They've made a commitment to the norms of the world they live in. Children are born not knowing what culture they've been born into, how the culture thinks, and what that culture thinks is important. Yet they are born with some built-in patterns of thinking that are universal. Since the late sixties, I've been interested in the extent to which you could cultivate the kind of thinking skills that only a few people use in the world today, by getting children to learn much more widely and much more fluently than most adults have. If you want to make a change, get the children to think differently.

This is one of the big reasons behind the \$100 laptop. Particularly in the Third World, where they value the idea of education much more, in a way that's very different from in the United States and Europe, there are many opportunities to see what can be done to help children.

People will attempt to spread our society's pop-culture technology to the Third World as the Third World gets wired up, so there's only a little window of time where you could possibly get children hooked into stuff which requires a lot more learning. It might not be possible, but I can't think of anything that I would rather work on. ☺

Please send questions and comments on this article to editors@ziffdavis.com.