Fred Brooks is known for his pioneering work in the visualization of molecular models. This year, at SIGGRAPH, he received the Allen Newell Award, which is given for career contributions that have breadth within computer science, or that bridge computer science and other disciplines. He is also the author of the widely read book, The Mythical Man-Month (1975, Addison-Wesley). Here he discusses tools of the present and future for scientific visualization, remotely held meetings and other applications. Dr. Brooks is Professor of Computer Science at the University of North Carolina, Chapel Hill.

A Conversation With Fred Brooks

Karen Frenkel: How widespread are the simulation- based design efforts going on today and and how important are they?

Fred Brooks: The most serious simulation-based design efforts underway today are sponsored by ARPA and are engaging various companies in American industry. The simulation is really crucial, and I think that the ability to link together many different simulation systems with a visualization system is going to be a paradigm of the future for lots of major design efforts—in heavy manufacturing for example. We already use simulation-based design in electrical circuits. People wouldn't think of building a circuit without running a SPICE simulation—or some equivalent simulation—for example. So I think that simulation as the basis for design—essentially providing the ability to build virtual prototypes—is a very important new trend. It is a major opportunity for the computing community to substantially help American industry.

What is the ultimate purpose of a visualization compared to a simulation?

Visualization has as its purpose communicating what the computer model is saying, or what the data is saying, to the mind. It is the mind-machine relationship. The simulation is essentially the process of generating something to say inside the machine. And so the simulation is playing out the computer model. The visualization helps the user to understand what consequences the model is generating and reporting as it does the simulation. So they're two different parts of the same process.

What are the differences between simulation tools and visualization tools?

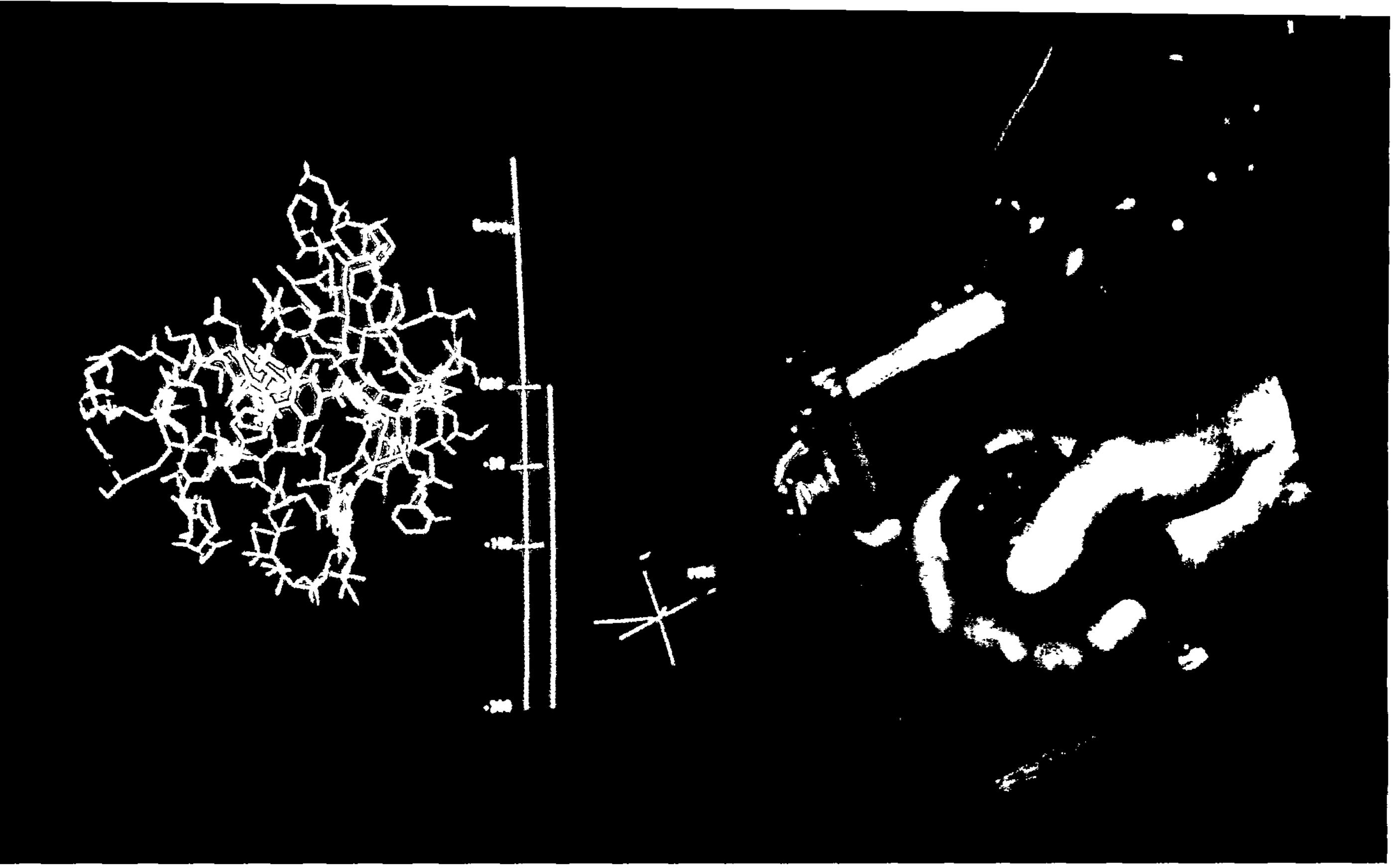
The two sets of tools have totally different purposes. Let's take molecular docking, in which you're concerned with how a drug interacts with a protein. The purpose of the simulation is to represent the physics and the chemistry and to calculate from them and from as close to first principles as you can afford to compute, exactly what the nature of the interaction would be under various postulated conditions. One of the powers of simulation is that you can investigate a lot of different conditions. You can explore large parameter spaces. Now I'm doing that for the purpose of understanding how these things will come together. And the simulation produces a lot of numbers—a lot of numbers! So the purpose of visualization is to make a lot of numbers comprehensible, so that the mind can grasp what happens if I use this drug instead of that drug, for example. How well does it dock into the protein? Where does it fail to fit properly? Why does it fail to fit properly? Is it a bad electrostatic charge? Or a bad hydrophilic relationship? Or just bad geometry? The purpose of visualization is to help me understand the details of what the simulation calculates. The visualization tools are inherently graphical, because their purpose is communicating to the mind from the machine model. The simulation tools are inherently not graphical, because their purpose is to work out the computer model.

What visualization concepts and techniques look particularly promising to you?

Three-dimensional interactive visualization, where the viewer is able to interact with the data rather than merely seeing a presentation of the data, is the most important and promising of the new concepts. This is being pioneered today in the journal Protein Science, where they publish a MacIntosh disk along with the printed text. It contains not merely figures but the data with which the viewer can see things other than what just what a picture can portray. Pictures move, you can select parts, and you do various interactive things with the data. I think that's an important new trend.

(Page opposite) Dr. Brooks demonstrates the GROPE experimental force display used in drug-protein docking studies.

Three-dimensional interactive visualization, where the viewer is able to interact with the data rather than merely seeing a presentation of the data, is the most important and promising of the new concepts.



Closer view of force display arm and molecule docking program.

Are you suggesting that electronic publishing can be considered a kind of new tool in computing and visualization?

Yes, because you don't just get the picture, you get the data and the tools to manipulate the data. You can make variations on the visualization to suit yourself.

Are there any other aspects of these tools that you consider important innovations?

In the area of scientific visualization and engineering-design visualization, the ability to feel forces, as well as seeing things, seems to me to be important. The availability of low-cost force displays seems to me to be an important innovation.

How might that manifest itself other than as a data glove with force feedback?

Right, a data glove is not such a hot gadget. You have a little handle that you hold and then you probe into the data holding the handle. And you feel the forces in three or more degrees of freedom as you work into your working volume.

That would be an important tool for molecular modeling.

Oh, I think it is going to be important in molecular modeling and in areas of manufacturing—planning assembly sequences—and probably in some areas of medicine. However, the forces that you're dealing with in surgery are very small and subtle, so that's more difficult.

What other fields are open to more use of visualization?

Long ago engineering design became the first area to justify commercial graphics on a large scale. Flight simulators, and vehicle simulators in general, has long been an area that had real-time interaction as a crucial component. That's been an active field for 40 years or more. Now we're seeing entertainment as a big field for all kinds of visualization. I think that science, entertainment, engineering design and increasingly, manufacturing planning are going to be important areas.

The people in some of those fields traditionally have not been dependent on those outside to make their tools. Is it a relatively new thing for computer scientists to come into another discipline, learn the application area, and then make the tools for people in that discipline?

What we see today is that computer science people increasingly make the hardware and the operating systems and the language compilers. But its still the people with an intimate knowledge of the application—sometimes working with a computer scientist and sometimes not—who develop the application programs. I think this works best when the application people work closely with computer scientists. I think the most effective cases have been teams.

Are you an advocate of teamwork among different disciplines?

Oh yes!

What kinds of tools work best for interdisciplinary purposes?

Well, in the work I've seen, I have been disappointed by the fact that the starting point was often the concept of the tool rather than the study of what people's habits and customs are when they collaborate. There seems to be very little published work on collaboration in the process of doing science, engineering, and design. The starting place has to be the study of what people have to do when they collaborate. So many of the efforts have been tool-driven or technology-driven, rather than need-driven. I think that may account for the slowness of their adoption.

Do we need special tools to enable people to have discussions about materials that everyone has seen remotely?

(Laughter) We fax copies of materials to one another and talk about them on the telephone. This we all do all the time. I think that is still going to be the most common technique. There

Excerpts from "The Computer Scientist as Toolsmith - II"

Presented by Fred Brooks on his receipt of the Allen Newell Award at ACM SIGGRAPH, July 27, 1994.

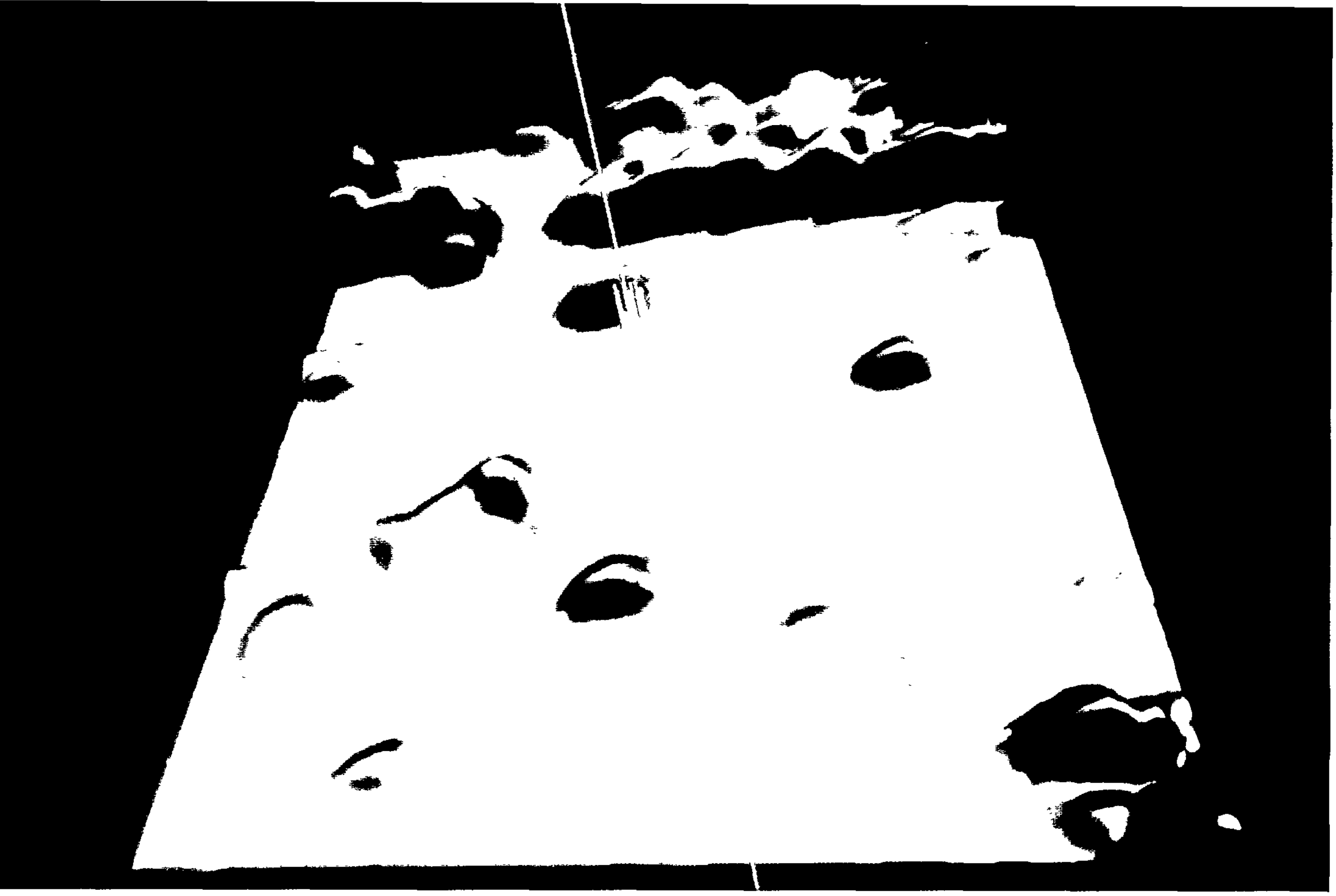
In contrast with many engineers who make houses, cars, medicines, clothing for human need and enjoyment, we make things that do not themselves meet human needs, but serve as tools in the meeting of needs. In a word, the computer scientist is a toolsmith—no more, but no less. It is an honorable calling.

If we perceive our role aright, we then see more clearly the proper criterion for success: a toolmaker succeeds as, and only as, the users of his tools succeed with his aid. However shining the blade, however jeweled the hilt, however perfect the heft, a sword is tested only by cutting. The wordsmith is successful whose clients die of old age.

If the computer scientist is a toolsmith, and if our delight is to fashion power tools and amplifiers for minds, we must partner with those who will use our tools, those whose intelligence we hope to amplify.

The magic of graphics, backed by megaflops of computer power, does indeed give us a creative medium of a totally new kind. We can sub-create worlds that work by their own laws, and we can immerse ourselves in them in ways that occasionally fool the mind. These worlds can show us new truth from our own world, through scientific modeling and visualization. They can show us new excellence, new beauty, flowing directly from our imaginations.

Without going into any detail, I would suggest that getting information from the machine into the head is the central task of computer graphics, which exploit our broad-bandwidth channel. Our other channels each have unique properties, however, and we must not neglect sound and haptics as ways into the subconscious parts of the mind. Likewise, in getting information from the mind back into the machine, one thing we may be sure of is that character strings are not usually the natural or right mechanism. We want to communicate as we do with other minds, by speaking commands, and by speaking, pointing, or moving to identify What? Where? How far?



A 20 nm colloidal gold particle being manipulated by the atomic force microscope (AFM) into a 150 nm gap previously patterned in a gold thin film. The yellow lines mark pushing events by the AFM tip against the particle. The indirect path indicates the sublety of particle manipulation.

is no substitute for one member of a team creating an initial draft or proposal which others then discuss. I do not observe very many cases in my life—doing design and doing science—in which people jointly create word-by-word or artifact-by-artifact as they go. I see somebody go through the dreariness of labor and the loneliness of thought to come up with a proposal, which others then take off from, contribute to, critique, modify, and maybe the net result is a totally new proposal or a totally new direction. But it seems to me that the collaborative process most often starts with a piece of individual work. And I think that we may have overemphasized the notion of sharing in the initial creation to an unrealistic extent.

So you don't foresee the need for special meeting tools?

Oh, yes, I do indeed. I'm on a National Research Council committee that meets jointly in Washington and on the West Coast. In front of us we have television screens so that we can see each other, and we share an electronic white board. That really facilitates such a meeting, and yet such a meeting is not at all the same as having everybody in the same room. Also, I teach on a two-way television link to as many as four other schools at the same time in which I can see the students and they can see me. We've done this now for seven or eight years in North Carolina, for advanced courses that are offered in only one of the schools. That's a pretty good substitution for everyone being in the same place. I display material and point to it, but in a teaching situation there's not modification of the material itself going on.

I want to return to our earlier discussion of computer scientists working in application areas in other fields. In that regard can you shed some light on the cross-disciplinary nature of your work? What advice can you offer?

First, you really need to form a close collaboration with the person who is going to be using the tool you're developing. A second point, which may surprise you, is that you'll almost never get it right the first time, and so it requires iterative development with the application person trying the tool and then coming back and saying "Here's what I like, and here's what I don't like." I think its very important that the application people begin by outlining their needs. At the same time, the computer scientist, who needs some familiarity with the applications field, must explain to the application person what the technology can do today that it couldn't do yesterday. Is that of any use to you? Can you think of any way of using that? Turn your imagination loose! So there is clearly a two-way process. But I'm stronger on the application, need-driven mode of development than on the technology-driven mode of development. It's also just fun to work with application people on important problems—to look over their shoulders while they explore how these marvelous natural things work. In the process you have to learn to speak the application language. You don't have to learn to do the application, but you have to learn its basic concepts and vocabulary.

I remember that people once thought there could be a problem in the mixing of scientists, for example, working on the human genome project, because computer scientists were used to focusing on their own problems, and geneticists on theirs, etc. Some computer scientists considered that kind of shared working situation almost servile because they wouldn't be the stars of the team. Is there a change in attitude? Is there more willingness now on the part of computer scientists to do applications work in other domains?

I don't know if there has been an attitudinal change over time or not. There have always been some computer scientists who felt that way. And there have always been some who enjoyed working on application problems. So I don't know that I have any way of estimating whether it is a trend or a shift in one way or another. It is typically the case that the people who make the best use of computer tools are the very ones who have been doing some computer application work by themselves without a computer scientist. They're much more apt to understand the computer craft than the computer scientist is to understand the application craft. Necessity may be driving more computer scientists to consider application areas now than in the 1960s or 1970s. I don't know whether there has actually been a positive change in attitude or not, but I'm trying to encourage one.