Frederick P. Brooks, Jr.

*Fred Brooks is the first recipient of the* **ACM Allen Newell Award***—an honor to be presented annually to an individual whose career contributions have bridged computer science and other disciplines. Brooks was honored for a breadth of career contributions within computer science and engineering and his interdisciplinary contributions to visualization methods for biochemistry. Here, we present his acceptance lecture delivered at SIGGRAPH 94.*

# *The Computer Scientist as* Toolsmith II

It is a special honor to receive an award named for Allen Newell. Allen was one of the fathers of computer science. He was especially important as a visionary and a leader in developing artificial intelligence (AI) as a subdiscipline, and in enunciating a vision for it.

What a man *is* is more important than what he does professionally, however, and it is Allen's humble, honorable, and self-giving character that makes it a double honor to be a Newell awardee. I am profoundly grateful to the awards committee.

Rather than talking about one particular research area, I should like to stay in the spirit of the Newell Award by sharing some lifetime reflections on the computer science enterprise, reflections which naturally reflect my convictions about the universe. The title and opening section of this talk were first formulated for a 1977 speech [1]. Let me reiterate the points, since many of you were barely born then.

In some quarters and at some times, computer graphics has been seen as a left-handed stepchild of computer science. Another view of computer science sees it as a discipline focused on problem-solving systems, and in this view computer graphics is very near the center of the discipline.

### A Discipline Misnamed

When our discipline was newborn, there was the usual perplexity as to its proper name. We at Chapel Hill, following, I believe, Allen Newell and Herb Simon, settled on "computer science" as our department's name. Now, with the benefit of three decades' hindsight, I believe that to have been a mistake. If we understand why, we will better understand our craft.

### What is a Science?

Webster says science is "a branch of study concerned with the observation and classification of facts, especially with the establishment and quantitative formulation of verifiable general laws." [2]

This puts it pretty well—a science is concerned with the *discovery* of facts and laws.

A folk adage of the academic profession says, "Any-thing which has to call itself a science isn't." By this criterion, physics, chemistry, geology, and astronomy may be sciences; political science, military science, social science, and computer science are not.

Perhaps the most pertinent distinction is that between scientific and engineering disciplines. That distinction lies not so much in the *activities* of the practitioners as in their *purposes*. A high-energy physicist may easily spend most of his time building his apparatus; a spacecraft engineer may easily spend most of his time studying the behavior of materials in vacuum. Nevertheless, the scientist *builds in order to study;* the engineer *studies in order to build.*

### What is our Discipline?

I submit that by any reasonable criterion the discipline we call "computer science" is in fact not a science but a *synthetic,* an engineering, discipline. We are concerned with *making things*, be they computers, algorithms, or software systems.

Unlike other engineering disciplines, much of our product is intangible: algorithms, programs, software systems. Heinz Zemanek has aptly defined computer science as "the engineering of abstract objects. [6]" Even when we build a computer, the computer scientist designs only the abstract properties—its architecture and implementation. Electrical, mechanical, and refrigeration engineers design the realization.

In contrast with many engineers who make houses, cars, medicines, and clothing for human need and enjoyment, we make things that do not themselves directly satisfy human needs, but which others use in making things that enrich human living. 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 tool succeed with his aid. However shining the blade, however jeweled the hilt, however perfect the heft, a sword is tested only by cutting. That swordsmith is successful whose clients die of old age.

### How can a Name Mislead Us?

If our discipline has been misnamed, so what? Surely *computer science* is a harmless conceit. What's in a name? Much. Our self-misnaming hastens various unhappy trends.

First, it implies that we accept a perceived pecking order that respects natural scientists highly and engineers less so, and that we seek to appropriate the higher station for ourselves. That is a self-serving gambit, hence dubious. It is also a risky gambit; in the case of some upstart social "sciences" the name is merely ludicrous and makes the practitioners look foolish. Moreover, the gambit is futile—we shall be respected for our accomplishments, not our titles.

Second, sciences legitimately take the discovery of facts and laws as a proper end in itself. A *new* fact, a *new* law is an accomplishment, worthy of publication. If we confuse ourselves with scientists, we come to take the invention (and publication) of endless varieties of computers, algorithms, and languages as a proper end. But in design, in contrast with science, novelty in itself has no merit. If we recognize our artifacts as tools, we test them by their usefulness and their costs, not their novelty.

Third, we tend to forget our users and their real problems, climbing into our ivory towers to dissect tractable abstractions of those problems, abstractions that may have left behind the essence of the real problem.

We talk to each other and write for each other in ever more esoteric vocabularies, until our journals become inaccessible even to our society members, and publication properly commands a higher price from the author in page charges than from the reader in subscription fees. So our writings even in their economics resemble garbage, for which the generator pays the collector.

**The scientist** *builds in order to study;*
**the engineer** *studies in order to build.*

This deadly trend already curses American mathematics; its cold chill can be felt in computer science. We are succumbing to the occupational illness of teachers diagnosed 2000 years ago by Jesus Christ: "You desire praise from one another. [John 5:44]"

Fourth, as we honor the more mathematical, abstract, and "scientific" parts of our subject more, and the practical parts less, we misdirect young and brilliant minds away from a body of challenging and important problems that are our peculiar domain, depriving these problems of the powerful attacks they deserve.

### Our Namers got the "Computer" Part Exactly Right

Some have wished that our discipline, and our professional society, were not named for a machine. I think Newell and Simon were exactly right on this point. The computer enables software to handle a world of complexity not previously accessible to those limited to hand techniques. It is this new world of complexity that is our peculiar domain.

Especially important for us are system design problems characterized by *arbitrary complexity*. Examples are the intricate demands upon operating systems, or knowledge webs, or computer networks. The arbitrariness is inherent—the requirements and constraints spring from a host of independent minds.

These problems scandalize and discourage those who approach them from backgrounds of mathematics and natural science, and for different reasons. Mathematicians are scandalized by the complexity—they like problems which can be simply formulated and readily abstracted, however difficult the solution. The four-color problem is a perfect example.

Physicists or biologists, on the other hand, are scandalized by the arbitrariness. Complexity is no stranger to them. The deeper the physicists dig, the more subtle and complex the structure of the "elementary" particles they find. But they keep digging, in full faith that the natural world is *not* arbitrary, that there is a unified and consistent underlying law if they can but find it.

No such assurance comforts the computer scientist. Arbitrary complexity is our lot, and here more than anywhere else we need the best minds of our discipline fashioning more powerful attacks on such problems.

It is too late to change our established name. Hence my purpose is not to propose a renaming, but to raise conscious mental defenses against the subconscious attitudes. The most important of these defenses are a continual focus on our users and a continual evaluation of our progress by their successes.

## The Gift of Subcreation

Making things has its glories and joys, and they are different from those of the mathematician and those of the scientist. Let us reflect together on these in a fundamental way.

The creation account in Genesis 1–2 is marvelously rich and subtle, and it can be read on many levels. I am not myself a seven-day creationist, but I take the account very seriously. It reports that our Maker gave humanity seven incredibly splendid "birth-day" gifts. Pondering the list, we see the satisfactions of our deepest longings and the provision of our greatest joys (see Figure 1). Here, I want to focus on the last, the gift of *work,* of the capability and the call to make things.

J.R.R. Tolkien, author of the epic *Lord of the Rings* trilogy, spent his life building a rich fantasy world with its own laws, species, languages, and geography. He calls this creativity the gift of *subcreation,* and he illuminates it in a poem peculiarly relevant to the graphicists' craft [5]:

> Although now long estranged,
> Man is not wholly lost nor wholly changed,
> Dis-graced he may be, yet is not de-throned,
> and keeps the rags of lordship once he owned:
> Man, Sub-creator, the refracted Light
> through whom is splintered from a single White
> to many hues, and endlessly combined
> in living shapes that move from mind to mind.
> Though all the crannies of the world we filled
> with Elves and Goblins, though we dared to build
> Gods and their houses out of dark and light,
> and sowed the seed of dragons—'twas our right
> (used or misused). That right has not decayed;
> we make still by the law in which we're made.

Tolkien applies this idea especially to the creation of fantasy, and fantasy worlds; I follow the English writer Dorothy Sayers in applying it to all human making [4].

A little reflection shows us that the power to make things, in imitation of our Maker, is a gift for our sake, not his. As he scornfully reminded the people of Israel, he doesn't need our creative powers:

"The cattle on a thousand hills are mine; if I were hungry, would I ask *you?*" [Psalms 50:12].

So we must conclude that the ability and the call to create are given to us to enrich our lives and to enable us to enrich each other.

## A Wholesome Evolution of Artificial Intelligence

Over the years since its beginning, the field of AI has made a wholesome evolution, which it is now time to observe and praise. In the beginning, the practice was primitive, but the rhetoric of the field echoed the builders of the Tower of Babel: "We will make machines that think; we will make Giant Brains." Just around the corner, given sufficient money and effort, were marvelous robots that could recognize visual patterns and spoken language, plan complex actions, answer sophisticated questions, and provide for all professionals the skills of the most expert.

**Figure 1.**
Seven birthday gifts to humanity on the occasion of Creation

- Life, and deathlessness
- Companionship with the Maker
- Friendship, especially marriage
- Children
- Nature, especially animals
- Freedom
- Creative work to do

A tremendous national investment has been made, over the course of more than three decades. Indeed, I would argue that too large a fraction of this country's public investment in computer science research went into AI, compared to other promising opportunities. More serious even than the diversion of dollars was the diversion of the very best computer science minds of a generation, and much of the efforts of the very best academic laboratories.

The by-products of this research effort have been impressive: new data structures and ways of representing knowledge, programming languages, families of computers. As for the main objectives, however, the field has accomplished surprisingly little for the time and the investment. One need look only at the present state of speech recognition and of handwrit-

*It is time to recognize that the* **original goals of AI were not merely extremely difficult,** *they were goals that, although glamorous and motivating,* **sent the discipline off in the wrong direction.**

ing recognition to see how far there is to go, despite how much work has been done.

At one time it looked as if at least the field of expert systems would prove a triumph, although many of the other goals were still elusive. Then came the rude awakening: somewhere in the neighborhood of 2,500 to 3,000 rules, rule bases become crashingly difficult to maintain as the world changes. Determining the consistency between new or changed rules and the rest of the base gets very hard, so hard as to put an effective upper limit on the usable size of the rule base. So today we have a useful expert system technology, with many examples of systems with a few hundred rules, but not the infinitely extendable tool originally dreamed of.

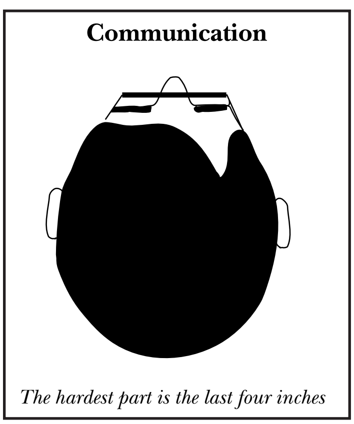



**Figure 2.** The communication problem

These years of experience give AI workers a deeper respect for the power of the human mind. The approach to AI systems has changed bit-by-bit, and now we hear the practitioners offering a "pilot's assistant," a "drilling advisor," or a "planning tool."

As real accomplishment has increased, the rhetoric has moderated. This evolution has been entirely wholesome. It recognizes a fundamental truth that was perhaps best articulated by Walt Kelly, creator of the comic strip *Pogo.* Albert Alligator has just depreciated the intelligence of Ole Bear, whose mental gifts are indeed limited. Ole Bear responds with a memorable line:

"Don't you go runnin' down my head-bone. They's some pretty fancy things goes on up there."

It is time to recognize that the original goals of AI were not merely extremely difficult, they were goals that, although glamorous and motivating, *sent the discipline off in the wrong direction.*

If indeed our objective is to build computer systems that solve very challenging problems, my thesis is that

**IA > AI**

that is, that *intelligence amplifying* systems can, at any given level of available systems technology, beat AI systems. That is, a machine *and* a mind can beat a mind-imitating machine working by itself.

Someday a computer may beat the world champion in chess. When that day comes, I should like to see the world champion equipped with a powerful and suitable IA chess tool, and then play against the AI system. I'll bet on the IA team.

Now the point of all this is that a different long-run goal aims our research in a different direction. Instead of continuing to dream that computers will replace minds, when we decide to harness the powers of the mind in mind-machine systems, we study how to couple the mind and the machine together with broad-band channels, an area of research dear to SIGGRAPH and one that has not yet received a small fraction of the attention given to AI research. The problems here are challenging and formidable, as the ad depicted in Figure 2 points out.

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 exploits our broadest-band 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 for certain 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?*

### The Toolsmith as Collaborator

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 intelligences we hope to amplify. Let me share with you some of our experiences in interdisciplinary collaboration at Chapel Hill over the last 30 years. It has been an exciting experience, and I commend it to you as a way of working. It also has some inherent costs, which one should intentionally decide whether to pay, and some inherent pitfalls.

### The Driving-Problem Approach

Let me begin with a paradoxical thesis:

*Hitching our research to someone else's driving prob-*

*lems, and solving those problems on the owners' terms, leads us to richer computer science research.*

This is a special case of the "down-is-up" paradox that governs so much of life, from marriage enrichment to career progress.

How can such a thing be so? How can working on the problems of another discipline, for the purpose of enhancing a collaborator, help me as a computer scientist? In many ways:

- It aims us at relevant problems, not just exercises or toy-scale problems.
- It keeps us honest about success and failure, so that we don't fool ourselves so easily.
- It makes us face the *whole* problem, not just the easy or mathematical parts. In computational geometry, for example, we can't avoid the cases of collinear point triples or coplanar point quadruples. We can't assume away ill-conditioned cases.
- Facing the whole problem in turn forces us to learn or develop new computer science, often in areas we otherwise never would have addressed.
- Besides all of that, it is just plain fun to look over the shoulders of those discovering how proteins work, or designing submarines, or fabricating on the nanometer scale.

In our Chapel Hill laboratory, our virtual reality team has been working with collaborators on the applications shown in Table 1. What specific computer science results, you might fairly ask, have you learned?

Table 2 shows some computer science results just from our work with molecular structure chemists. A pretty side effect is that the polygon simplification algorithm developed for molecular surfaces turned out to save the day on real-time visualization of a 350,000 polygon model of part of a submarine.

Gleaned from a video depicting some recent experiments, Figure 3 shows physics graduate student Michael Falvo tapping a gold ball into the intended gap in a circuit using an atomic force microscope. Figure 4 shows a few frames in the rearrangement of the parts of a tobacco mosaic virus by tapping with the probe of an atomic force microscope.

## The Costs of Collaboration

There are real costs associated with any professional collaboration, and interdisciplinary collaborations have some unique costs. I find that our teams spend about a quarter of our professional effort on routine work that supports our collaborators but does not advance our joint researches, much less the computer-science part of the research. A chemist needs a special illustration made for a paper or for a textbook cover. A submarine designer needs a special technical demo for his funding agency or management. These we gladly do. The shoe is often on the other foot. Our chemist collaborators spend hours tutoring our graduate students in the elements of protein structures and guiding them through hands-on exercises with brass and plastic models.

All collaborations require time in planning and communicating among the senior scientists. This work cannot be delegated—only the bosses on the two sides can do it.

Finally, it is necessary for our faculty and students to spend some of their time learning protein chemistry, surface physics, radiology, or architectural design. Our Ph.D. students often take introductory courses in the using disciplines, and they always take reading courses from our collaborators to prepare them for their dissertation work. One need not become expert in the partner's field, of course, but one does need to learn the basic principles, the vocabulary, and the partner's research objectives.

## Terms of Collaboration

No two partnerships are alike, and with good will many different arrangements can be made to work. We have found some simple principles to help our intellectual interdisciplinary collaborations. One of the most helpful is for neither partner to be a contractor for the other—each raises his own support. This ensures that there are no artificial strings tying together a collaboration after one partner has found it to be no longer worth the

**Table 1.** Virtual reality driving problems

| |
|---|
| •Medical imaging and reconstruction |
| •Radiation treatment planning |
| •Molecular structure |
| •Control of scanning probe microscopes |
| •Design of buildings and submarine spaces |
| •Debriefing of fighter pilot tactical exercises |

**Table 2.** Some CS results from molecular graphics driving problems

| |
|---|
| •Don't overload manual devices with multiple functions |
| •Force displays can make molecular docking up to two times faster |
| •New linear-time parallel alpha-hull algorithm |
| •New polyhedron simplification algorithm |

investment. A collaboration works only when everybody wins.

Two of our criteria for success in a tool are:

- It must be so *easy* to use that a full professor *can* use it, and
- It must be so *productive* that full professors *will* use it.

Ph.D. students can and will use any crummy tools that inch their theses along. We consider our tool-building collaborations to be a success when the senior collaborating scientists use our tools for their personal work. Building tools that satisfy these demanding criteria requires close work with the collaborators, iterating on the definition of what is useful.

What about credit? On a championship team, everybody gets a ring. On a winning collaboration, there is plenty of credit to go around. I have never known credit to be a problem in an otherwise successful collaboration.

## Entertaining Doubts

Now let us turn especially to computer graphics and SIGGRAPH. No part of computer science is more wonderful nor more fun. Our conference is a joyous celebration of a steady succession of advances in hardware, software, and concepts of use. We have much to celebrate.

Nevertheless, I would lift a challenge before us. In a recent interview, Dan Goldin, the administrator of NASA, said, "I'm not worried about the space program. I'm worried about America. Our nation has become a nation of consumption. Entertainment and recreation are the most important things for the future. God help us!"

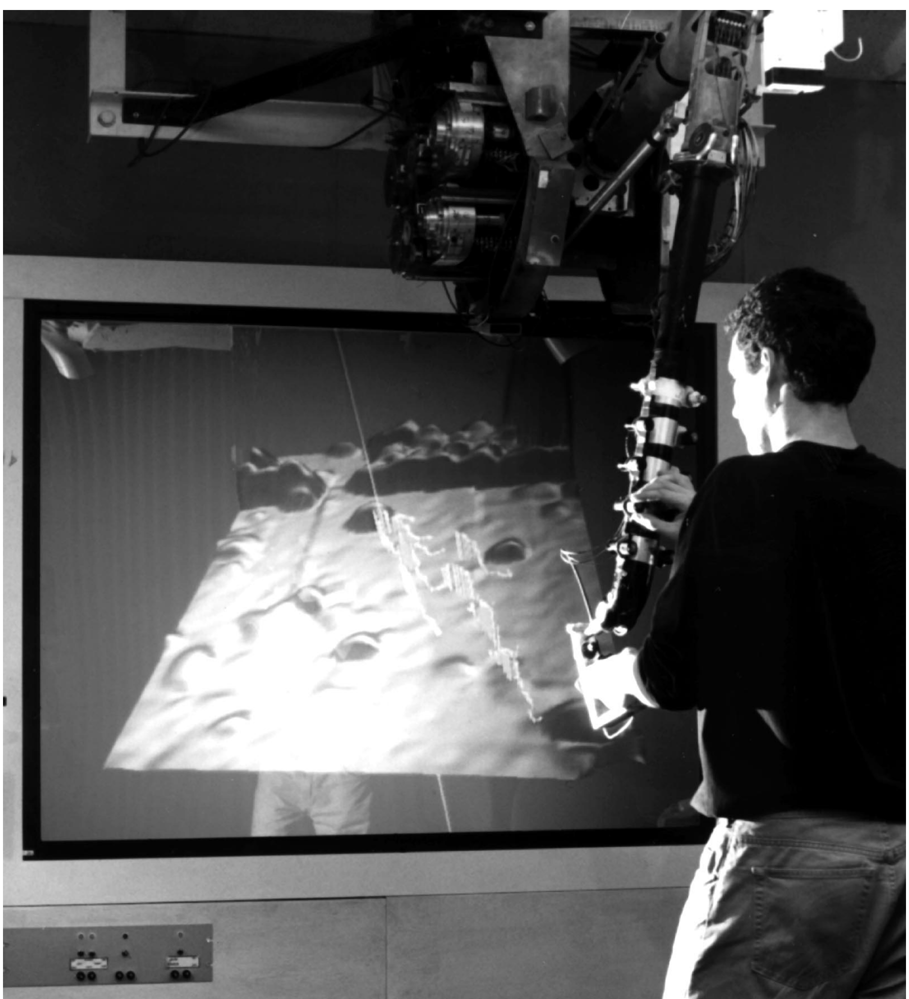


**Figure 3.** Using an atomic force microscope, Michael Falvo taps a gold ball into the desired gap in a circuit. (The entire field of his view is 1 nm.)

I share Goldin's concern. Rome rotted from the inside when its people became interested in nothing but bread and circuses. Let us consider just the one aspect of American entertainment and recreation that is especially pertinent for SIGGRAPH—TV.

At a recent college commencement, Mother Teresa received an honorary degree and the crowd applauded politely. Then, last on the list of honorees, came the one the crowd had come to see— Meryl Streep. The crowd exploded in applause [3]. Miss Streep is excellent at her craft, but what do we value? I am continually surprised at the warm congratulations we receive when the work of our laboratory appears on national TV, as if the exposure had somehow enhanced or validated the work. The O. J. Simpson trial showed us how fame feeds on itself. Media professor Neil Postman, in his book, *Amusing Ourselves to Death,* details how "Our politics, news, religion, [and] education have [become] adjuncts of show business. [3]"

As a recreational medium, TV has an unprecedented power that we can professionally appreciate—it is *visual.* But the medium is inherently *passive,* a shortcoming that drives many of us here to work on interactive media. TV is inherently *non-social,* and we know

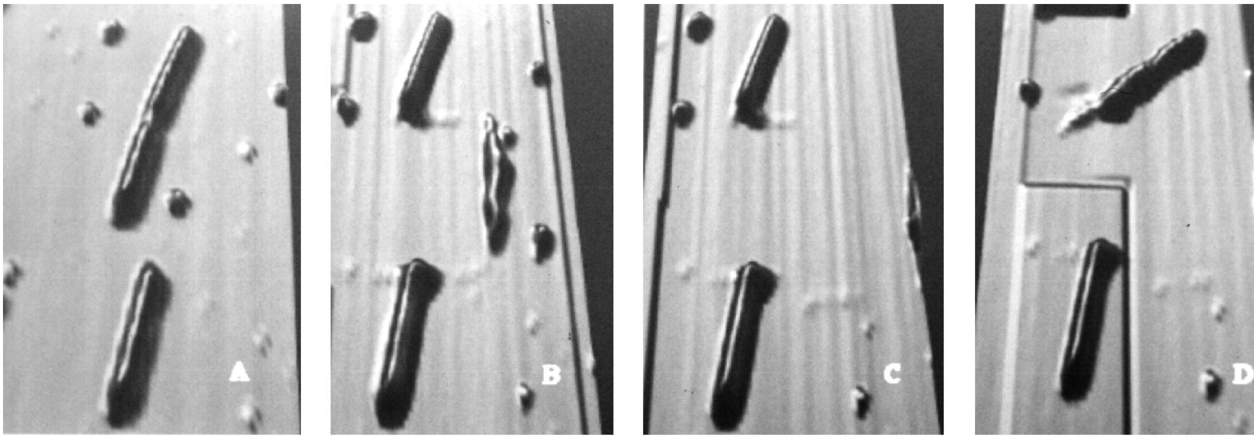
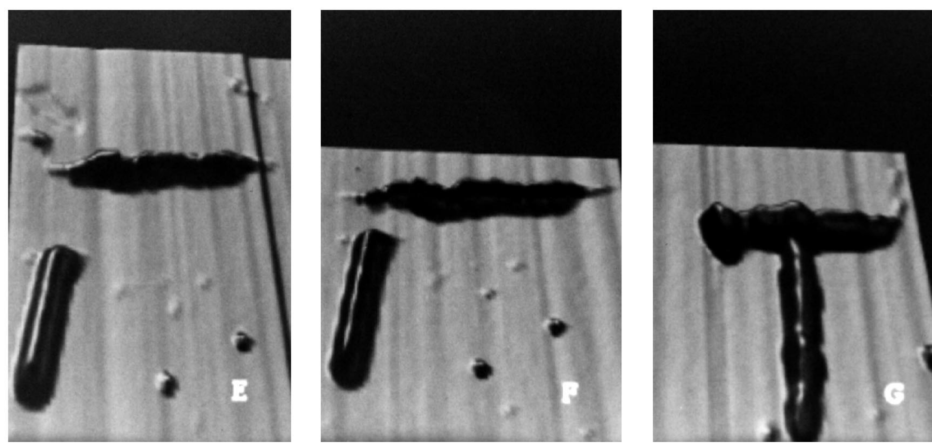



**Figure 4.** Rearranging parts of a tobacco mosaic virus by tapping with the probe of an atomic force microscope. (The scene is about 1 nm across.)




the value of interacting with people. And as practiced in America, TV is *frantic*—exciting but not refreshing, with the average cut lasting 3-1/2 seconds [3].

Minow once characterized American TV as a "vast wasteland." Today it has become vaster, with many more channels, and more to come. It has also become more desolate. I fear we rarely stop to contemplate how really bad it is.

The ancient Greeks asked of any aspect of life, Is it true? Is it beautiful? Is it good?

TV is not a bastion of truth. Because of TV, we now have to teach toddlers a lesson we once could long defer—that people lie, especially people selling things. Even more seriously, the program content teaches wrong implicit lessons about life—it avoids teaching about the sorrow, loss, and emptiness that come with any death, about the joys of growing old together, about the delight of raising children.

TV fails the beauty test. Although the cinematography is frequently very skillful, the overall effect is ugliness — bleak slumscapes, ugly violence, and endless car chases.

TV is only occasionally good. The voracious appetite for material means mediocre dramas. The characters are rarely people we should like to have as friends, quite unlike, for example, the people in Neville Shute's novels. Only rarely would we want our children to take TV characters as their role models.

On a late-life occasion honoring the inventor of the vacuum tube, Lee DeForest, he remarked on how the tube had made radio possible, and he sadly commented, "This is DeForest's prime evil." Today he would have a new candidate.

"What *did* people do before TV?" How did we recreate ourselves?

- People *visited* with each other.
- People *made* quilts, inventions, music, games.
- People *read,* letting their own imaginations furnish the pictures.
- People *played* sports, instead of mostly watching them.
- People *observed* nature, rather than pictures of nature.

Well, what has all of this to do with SIGGRAPH? Quite a bit; SIGGRAPH also worships TV and its fame. Nowhere is this more evident than in the Electronic Theatre. Year by year we increasingly choose what to honor by the standards of the TV culture. It is increasingly an Electronic *Theatre,* rather than a showcase of computer graphics. We are treated to luminous dancers, bogus lip-synched music, and cheap distortions of 2D video images of the real world.

Every year there are wonderful exceptions, from "Luxo, Jr." to the "Devil's Mine Ride," but I am struck that so often I can only marvel at what has been accomplished, rather than also delighting in it.

The same questions have to be raised about the Art Show:

- Where is sheer beauty? Isn't that what art is about?
- Where is delight that we can share with the artist?
- Have we abandoned art as subcreation for each other's enrichment, in favor of an art of self-exorcism, art as primal scream?

### What Can We Computer Graphicists Do?

The magic of graphics, backed by megaflops of computer power, does indeed give us a creative medium of a totally new kind. We can subcreate worlds that work by their own laws; we can immerse ourselves in these new worlds 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.

What comes out of a human imagination can be achingly beautiful or painfully ugly, deeply true or deeply false, wonderfully good or horribly evil. As Jesus said, what comes out depends upon the condition of the heart itself [Matthew 15:18]. If we would have our creations be true, beautiful, and good, we have to attend to our hearts.

As the Apostle Paul put it [Philippians 4:8]:

*"Fill your minds* with those things that are good and deserve praise; things that are

- true,
- noble,
- right,
- pure,
- lovely, and
- honorable."

**Acknowledgments**

The Newell Award is for lifetime breadth of work, and especially interdisciplinary work. It therefore implicitly recognizes my many collaborators. I want to make explicit my special thanks to those who have colabored with me through several decades.

- Everything—Nancy Greenwood Brooks
- Computer architecture—Gerrit Blaauw, Richard Case, John Cocke, John Fairclough
- Graphics—William Wright, Henry Fuchs, Michael Pique, David and Jane Richardson
- Building a computer science department— Peter Calingaert, Henry Fuchs, Stephen Pizer, Donald Stanat, Stephen Weiss.

**References**

1. Brooks, F.P. The computer scientist as toolsmith—Studies in interactive computer graphics. In *Information Processing 77, Proceedings of IFIP Congress 77,* B. Gilchrist, Ed. North-Holland, Toronto, 1977, 625–634.
2. Neilson, W.A., Ed. *Webster's New International Dictionary of the English Language, 2d ed., Unabridged.* G.C. Merriam, Springfield, Mass., 1960.
3. Postman, N. *Amusing Ourselves to Death: Public Discourse in the Age of Show Business* Penguin, New York, 1985
4. Sayers, D.L. *The Mind of the Maker* Harcourt, Brace, New York, 1941, Chap. III.
5. Tolkien, J.R.R. *Tree and Leaf.* George Unwin, London, 1964; American edition, Houghton Mifflin, The Riverside Press, Boston, 1965, p. 54.
6. Zemanek, H. Was ist informatik? *Elektron. Rechenanlagen 13,* 4 (Aug. 1971), 157–171.

**About the Author:**

**FREDERICK P. BROOKS, JR.** is Kenan Professor of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175; email: brooks@cs.unc.edu