WHAT THINGS LOOK LIKE

Ivan Edward Sutherland

PART 1: THE PAST

Andrew Carnegie contributed to my college education by endowing the Carnegie Institute of Technology, now Carnegie Mellon University. Andrew Carnegie was, in his time, one of the richest men in the world. He became famous for his support of libraries and church organs. He gave matching funds, giving only half the required amount. Local contribution of the other half enhanced the value of Carnegie's gift. Some say that when Andrew Carnegie got to heaven he had only enough gold left for half a halo and sought matching funds.

Carnegie's motto was "My Heart is in the Work." The motto suggests that the work came first, and then the heart got in. I like the motto, but I see it the other way around. When I work on things I love, my work warms my heart.

The best advice I ever got came from my son, Dean. I was asked to seek the presidency of a respected university. Dean asked me if the job seemed like fun to me—which it most certainly did not. I declined the opportunity, taking Dean's superb advice: "Don't do it, Dad, you're no good at things you don't think are fun."

I've always loved my work and so, for me, work has mostly been play. Now I'm old enough to retire, but I go on working because my heart is still in the work. Or rather, I am still able to find work that warms my heart. In a few minutes I shall say a little about the work that now warms my heart.

WHY I LOVE MY WORK

I was destined from birth to be an engineer. My mother was a teacher whose main goal was the education of her children. She taught me diction, insisting that I pronounce every syllable of the words I spoke. She listened to me recite poetry and made sure the meaning came through as well as the poetic meter and the rhymes. I still get pleasure from repeating some of the poetry I learned long ago. Thank you, Mother, for the legacy you gave me.

Mother hated to drive. After weeks of my pestering her, she drove me to see the Brooklyn Bridge. I remember vividly walking with her over its span. On another occasion she drove all the way through New York City to Bell Telephone Laboratories, taking her two boys to visit Claude Shannon. We saw his maze-solving mouse. It used relay logic because suitable transistors had yet to be invented. About ten years later, in 1961, I asked Shannon to supervise my PhD thesis. He agreed and still recalled the earlier visit of two young boys. Thank you, Mother, for taking us to visit Claude Elwood Shannon who, in 1985, accepted the first Kyoto Prize in Basic Sciences.

My father was a superb engineer. He had earned a PhD in Civil Engineering from the University of London in the 1920s. He had a giant body of practical knowledge that was much more important than his formal education. He had a “feel” for mechanical things: how they worked or why they didn’t work. I learned the value of his practical knowledge by watching him fix things. I remember learning from him how the float in a toilet turns off the water when the tank is full. I remember learning from him how the cylinders of a steam locomotive drive its wheels. I remember learning from him about three-phase electricity before I had enough mathematics to understand its details. He knew the math, but he had something much more important, a deep understanding of what the math was for. It’s not enough to solve equations, one must know what equations to solve and what the solutions mean. Thank you, Father, for the understanding you shared with me.

My parents conspired to teach their boys mental arithmetic. On long auto trips we’d often play the “I’m thinking of a number” game. “I’m thinking of a number. If you square my number and add two you get eighteen. What is my number?” Endless such questions, increasingly difficult as we matured, not only kept us out of mischief but also taught us algebra years before we learned it at school.

I had two favorite childhood toys. One was a set of Tinkertoys, wooden sticks and spools that I could assemble into moving machines. The length of the sticks and the angles of the holes in the spools encouraged using triangles for strength. A fine set of plans suggested how to assemble complex structures. I remember Mother insisting that each of my structures must match its plan. I was six or seven years old at the time. My set kept expanding as mother added more parts. I’ve lost track of where my set of Tinkertoys is now.

My other favorite toy was a set of old German building blocks that we called the Richter blocks. Although they appear to be made of stone, I later learned that they are a composite of sand and fine powder held together with polymerized linseed oil. Their detailed plans showed, layer by layer, how to build amazing buildings. I remember mother teaching me how to read the plans. Long before I learned to read English, I could read Richter block plans and build the buildings. I still have my blocks and the plans.

My school required protective paper covers on our textbooks. It was up to the family to provide the covers. All my friends had purchased book covers from Harvard or Yale, or some other prestigious university. Mother argued that we couldn't afford such elegant covers, and so she covered my textbooks with blueprints from my father's work. The blueprints showed the inside of the dams he built, where the turbines sat and how water got to them. I understood real blueprints before I got to high school.

With help from my father I built the gantry crane pictured here in an article from the school newspaper (Fig.1). A year later I built this model of a steam shovel, shown here with me in my Morrison Knudsen hard hat (Fig.2). I recently had a tour of the Vigor shipyard on Swan Island near Portland. There I saw a crane similar to my model, but able to lift 600 tons. The shipyard gave me a new hard hat as part of the tour. The hard hat and the crane reminded me of 60 years past when I savored such visits with my father.

GEOMETRY

Looking back now, I realize that from a young age I saw both plans and reality. I converted Tinkertoy plans into Tinkertoy structures. I converted Richter block plans into Richter block buildings. I learned to read plans for my father's dams and often got to visit the dams. The plan is not the reality; it only models reality.

Engineering depends on such models of reality. Engineering models are often drawings. Sometimes they are scale models. Sometimes they are mathematical equations. Modeling is the most important part of engineering. When our models are faithful to reality, they serve us well, and airplanes fly, buildings stand, and bridges carry traffic. When our models fail to describe reality, disasters happen. Japan recently suffered serious damage from a flawed model of ocean behavior.

Geometry has always been an essential part of my thinking. Both model and reality share the same geometry. Tinkertoy objects have the same geometry as their plans. Richter block buildings have the same geometry as their plans. Things fit because their geometry is right. The role of geometry in mechanical engineering is clear. But I am an Electrical Engineer. Where's the geometry in Electrical Engineering?

I evolved my own simple geometric model of electric circuits. In my simple geometric model force stands for current, and position or length stand for voltage. In my simple geometric model a spring stands for a resistor. When you apply a force, which stands for a current, to a spring by pulling on it, the spring changes length, which stands for the voltage on the resistor. To lengthen a spring requires applying force. The equation of Ohm's law that relates current and voltage in a resistor is the same as the equation that relates force and elongation in a spring. My simple geometric model is faithful to reality.

I also have a simple geometric model for transistors and for capacitors. In my simple geometric model, a monk pulling on a bell-rope that hangs through the ceiling stands for a transistor and a shock absorber from a car stands for a capacitor. The details of these models are quite technical, and so I'll leave explaining them for another day. It is enough to say here that my simple geometric model of transistor action and capacitor behavior is faithful to reality.

My simple geometric model is useful to me. It lets me "see" how a circuit works because I can imagine the motion of my simple geometric model in my "mind's eye." My model gives me a "feel" for circuits. It gives me a picture to supplement my mathematics. The mathematics tells me the accurate answer, but my geometric model tells me what mathematics to use.

COMPUTER DRAWINGS

Carnegie Tech had a required course in Engineering Drawing. It was required for two reasons. First, it taught how to read engineering drawings, a skill I already had from the blueprints protecting my schoolbooks. Second, it taught how to make beautiful drawings with pencil on paper. I hated the drawing part because it required manual skill and patience I found hard to muster. Worst of all, my eraser left an ugly mark on my drawing paper and left eraser dust everywhere.

Even then I wanted computer help. However, in those days a computer was the size of a room and very expensive to use. "On line" computer use had not yet been invented; "batch processing" was the common way to use computers. A user left a deck of punched cards for an operator to present to the computer. Later, after a wait of at least a few hours, the user picked up a large pile of paper printed entirely in upper case letters.

In 1960 I went to MIT as a graduate student. There I found a few computers with display screens that might be able to draw pictures. I had the first glimmer of making engineering drawings on a computer. Could these computers make the lines straight? Could these computers erase things without leaving ugly marks? No, the computers on the MIT campus then were too small for the task.

I had a summer job at MIT's Lincoln Laboratory, separated from the MIT campus. There I found the most powerful computer in the world, the TX-2. It had a screen that could display drawings. It had a light pen that could serve as a stylus. It had a huge memory, about 1/4 of a megabyte. And it was very fast: it did 80,000 operations per second. Although it was then the most powerful computer in the world, by today's standards it was primitive. Lincoln

Laboratory built it to learn how thousands of transistors would behave in a large computer.

Wes Clark designed the TX-2. To me, a mere graduate student, he seemed very senior and slightly scary. Wes and I later became good friends and so we remain to this day. Many years later he told me that he had built TX-2 especially for me in spite of not knowing who I was when he built it. So it seemed. Sketchpad needed and used all the power of TX-2. Unlike the batch processing computers of the time, TX-2 served each user individually for hours at a time. I needed that access to experiment with and improve Sketchpad.

At that time no one knew what computer drawings should do. Should they be a better version of pencil and paper drawings? Should computer drawings differ from paper drawings, and if so how? At the time I failed to ask such questions. I just wanted to make nice pictures on the screen. I learned by doing. Just as a baby must flex its limb muscles to learn how arms and legs work, so science must flex its intellectual muscles to learn how physical things work. I now know that this learn-by-doing process is a valuable activity. We call it “research.”

Did anybody notice that I used the verb “do” rather than the verb “be” when I asked what computer drawings should do? Sketchpad showed that computer drawings are much more than paper drawings because they *do*, they don’t merely *be*. Computer drawings can move, but paper drawings stay fixed. Parts of a computer drawing can stick together and remain attached, but paper drawings are just dirt on paper. Any meaning in a paper drawing lies strictly in the mind of a person. Computer drawings can have intrinsic meaning. That’s why I said “do” rather than “be.”

TX-2 had to work hard to make a line appear on the screen. All the screen itself could do was flash a single dot of light at any of about a million different places on a small screen. To make a line, TX-2 had to flash separate dots all along the line. Because the screen had no memory, TX-2 had to flash the same dots over and over and over to keep the line visible on the screen. A line was no longer a thing; a line became a process. A line was no longer a noun idea, a thing put there to stay. Instead a line was a repeated process, a verb idea. TX-2 didn’t *draw* lines, it *did* lines, and it did them over and over and over. Erasing was easy; if the process stopped, the line vanished. I had found how computer drawings behave rather than just what they look like.

I made a movie that was widely shown and remains on the web today. The drawings in the movie change shape. It’s well to remember that in those days only a handful of people had ever watched a computer screen. The idea of any computer drawing was novel, let alone a moving drawing.

Today computer drawings serve two purposes. Some of them serve, like paper drawings, only to offer knowledge or fun to people. Computer drawings for art and entertainment are now everywhere. Computer output drawings from scientific and engineering calculations are common. Such drawings bring understanding to people but do nothing for the computer itself. Such drawings are computer output. They let people see what things look like.

The second purpose of computer drawings is to define facts for computers. Drawings for computer-aided design are different from paper drawings. We expect computers to simulate the circuits we draw for them. We expect computers to calculate the behavior of linkages we draw for them. We expect computers to estimate the weight or strength of structures we draw for them. Such drawings are computer input. They let computers see what things look like.

In 1975 I published “A Characterization of Ten Hidden Surface Algorithms” with co-authors Bob Sproull and Bob Schumacker. It was my last effort in computer graphics. We examined all ten of the known algorithms for making solid-looking pictures. We discovered that all ten algorithms involved sorting. I realized that to make better pictures by computer would require ever better methods for sorting. Better sorting looked boring to me. Better sorting failed to warm my heart.

PART 2: THE FUTURE

In 1976 I joined Carver Mead on the faculty at Caltech. A revolution in electronics was underway. Carver had a simple way of thinking about the design of integrated circuit chips. Integrated circuit chips do logic with a geometric pattern of transistors and wires. I found the geometry fascinating. Integrated circuit chips are electrical circuits made of transistors and wires. I found the circuits fascinating. Integrated circuit chips do logic. I found the logic fascinating. All three together form a deliciously complicated design challenge that I found irresistible. A new kind of work warmed my heart.

Carver Mead and Linn Conway wrote a classic book on Very Large Scale Integrated (VLSI) circuits. It’s called “Introduction to VLSI Systems,” but it is best known merely as Mead and Conway. Carver Mead and I were colleagues at Caltech. Lynn Conway worked for my older brother, Bert Sutherland at the Xerox Palo Alto Research Center (PARC). Mead and Conway dedicated their book to Bert.

Xerox PARC invented a multi-project chip. The simple idea was to share the big cost of making a chip among a large number of small projects. Soon after, the US government funded a fabrication service, called MOSIS, to build multi-project chips designed by students. Overnight, it seemed, about 100 courses in integrated circuit design appeared.

That flowering of knowledge gave industry the engineers needed to fuel an explosive growth of integrated circuit technology.

I've now been in two major paradigm shifts. The first involved Sketchpad changing how people could use computers for pictures. I never intended to cause such a change. The change was an accidental result of things I did merely because they were fun. While I deeply appreciate the honor of the Kyoto Prize, I never sought such distinction.

The second paradigm shift involved integrated circuits. I never sought to enlarge the flow of engineering talent skilled in integrated circuit design. Others may have had that goal; I think Lynn Conway taught the teachers of integrated circuit courses in full knowledge of how important that would be. I merely did what seemed technically most interesting day by day.

I'd like to see a third major paradigm shift. For me this is a very different goal than I have ever before sought. I have never before sought paradigm shift. In the past I focused on better technology without thinking about wider adoption. Paradigm shift happened naturally as a result of better technology.

SELF-TIMING

In 1988 I published "Micropipelines" showing how to build a pipeline of self-timed parts. The idea is very simple. Let each part of a digital system do its job at its own pace. Self-timing is a radical idea because today's digital systems nearly all get timing from an external rhythm called the "clock."

Digital systems today nearly all quantize time into tiny "clock periods" according to the beat of a rhythmic external clock. The rhythmic clock is useful because it lets the designer know exactly which logic steps will occur in each and every clock period. The designer can check that everything required before each step already happened in previous steps. Clock periods simplify the logic design task.

On the other hand, clock periods complicate the electrical design of a chip. Each and every logic operation must fit into the chosen clock period. Simple logic that might be faster must wait. Difficult logic tasks that might be slower must hurry. Uniform delay costs design effort and wastes energy. Worst of all, reliable delivery of the clock itself requires design effort and wastes electrical energy.

CHANGING A PARADIGM

The clocked design paradigm, though widely used, looks to me like an ultimate dead end because of both a practical flaw and a fundamental flaw. The practical flaw is that the clocked paradigm entangles three hard design problems: logic design, electrical design, and geometric design. Any change in the geometry or the logic of a chip changes the number, length or placement of its wires and thus affects its electrical design. Any change in the logical or electrical design of a chip changes the size of its parts and thus affects its geometric design. Finally, any change in electrical or geometric design may retard the design enough to exceed the clock period and thus force changes in its logic. Design of large clocked chips is very expensive because their clocked paradigm entangles three hard design tasks. The entanglement appears most vividly during a process called “timing closure” in which each and every signal must be forced to fit into the fixed clock period.

I must offer an allegory to explain the fundamental flaw in clocked design. Many people watched NASA’s recent Mars probe, Curiosity, land on Mars. Mars was so far distant that light took thirteen minutes to get from Mars to Earth. It was impossible from Earth to see Curiosity land; instead we saw a delayed report that Curiosity had successfully landed thirteen minutes earlier. No one suggested that a pilot on earth could help Curiosity land. Curiosity had to be self-timed because of the thirteen-minute communication delay.

Each new family of integrated circuits has smaller transistors and finer wires. Smaller transistors go faster but finer wires carry signals at the same or a slower speed, a speed much slower than the speed of light, by the way. The smaller transistors need more precise “simultaneous” delivery of clock signals, but the finer wires make “simultaneous” harder to achieve. Today clocks can be “simultaneous” over only a small portion of a modern chip. The clocked paradigm can no longer apply to a whole chip.

The self-timed paradigm seems preferable to me. Self-timing offers modularity because separate modules are truly separate. Separate modules need not use a common clock; each can proceed at its own best pace. Self-timing untangles the three design tasks. The correctness of logic is independent of geometry and far less dependent on electrical design because each part will wait for the data it needs. Geometric changes that lengthen or shorten wires leave logic correct, changing only how long tasks take. Electrical changes or manufacturing variation likewise leave logic correct, changing only how long tasks take. Finally, because self-timing recognizes the fundamental relationship between time and space caused by communication delay, self-timed systems can scale to any size.

I want self-timing, rarely used today, to be a useful alternative for the design of digital systems. I believe that using self-timing offers big rewards. I seek to change the paradigm by which we do digital design.

WHAT SHALL I DO?

To cause a paradigm shift I face tasks totally unfamiliar to me. If I think in positive terms I ask these questions: How should I measure success? What can I do to hasten change? Is there a leverage point? Is there something simple for me to do that will lead others to see what already seems obvious to me?

If I think in negative terms I ask: Is the clocked paradigm too entrenched ever to change? Is my paradigm shift even possible? Is my goal unrealistic? How can I tell if I have a reasonable goal?

My real problem is to find what warms my heart. Do I really care about paradigm shift? Should I make change itself be my goal? Or should I, as I've done in the past, leave change to others and just do the work that I love?

I say all this here for two reasons. First, I need to solve my dilemma of what to do with the rest of my life. I hope that writing and talking about my dilemma will help me resolve it. My second reason for saying all this here is to let others know that even Kyoto Prize recipients face dilemmas of purpose, or at least to know that one Kyoto Prize recipient faces a dilemma of purpose. I thank the Inamori Foundation for this opportunity to help me find what warms my heart.

AN IMAGE OF THE FUTURE

I know three important things about this particular dilemma. First, there may be a leverage point. It may be possible to build a self-timed integrated circuit chip flexible enough to serve a wide variety of uses. Like my toy blocks, it could have many logic parts for a user to layer together to build a logic structure. It should remember the user's connections just as gravity kept my building blocks in place. Such a device already exists. It is a Field Programmable Gate Array or "FPGA."

Although clocked FPGAs are in widespread use, the existing clocked designs limit users to building clocked systems. Clocked FPGAs are a popular way to try out new ideas for clocked designs. A new self-timed FPGA would allow users to try out new ideas for self-timed designs.

A self-timed FPGA may be a leverage point because it will permit widespread use of self-timed circuits by many people. It could be the basis for thousands of experiments in self-timing. It could serve as a training ground for thousands of students. It could be the basis for new courses in self-timing. A self-timed FPGA chip may be the fertile soil in which the self-timed paradigm may flower.

The second thing I know is that building a self-timed FPGA chip will be fun. It will be fun because the design task it presents is hard enough to challenge my skill, but easy enough to finish within my lifetime. Most important, it affords me the chance to work with some of the best minds in my field. This project will let me stand proudly shoulder-to-shoulder with brilliant colleagues.

The third thing I know is the impossibility of forcing anyone to change. I can't force other people to use my favorite ideas. I can't force a paradigm shift, but I might enable one. The best I can hope for is that giving others easy access to self-timing may let them recognize its value. The paradigm shift will either happen or not, only time will tell.

I must, as I have done in the past, focus on the technical work because it is the technical work that warms my heart. I must avoid to worry about whether or not a paradigm will shift. It is enough that I have a likely leverage point, a fine technical challenge and apt colleagues. Let history decide whether a paradigm shifts as a result of our efforts. My heart is in the work, or as I prefer to say, this work warms my heart.

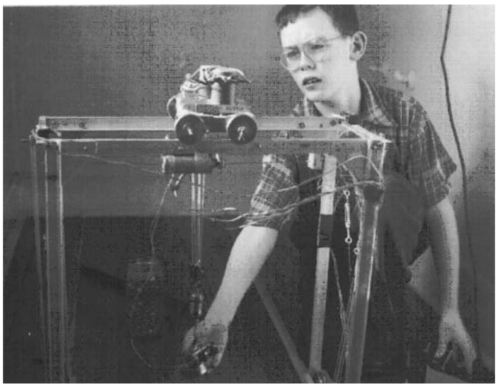

Fig. 1

Fig. 2