

題名	ものの見え方
Title	What Things Look Like
著者名	アイバン・エドワード・サザランド
Author(s)	Ivan Edward Sutherland
言語 Language	日本語・英語 Japanese, English
書名	稲盛財団：京都賞と助成金
Book title	Inamori Foundation: Kyoto Prize & Inamori Grants
受賞回	28
受賞年度	2012
出版者	公益財団法人 稲盛財団
Publisher	Inamori Foundation
発行日 Issue Date	9/30/2013
開始ページ Start page	104
終了ページ End page	130
ISBN	978-4-900663-28-2

ものの見え方

アイバン・エドワード・サザランド

第一部：これまでのこと

私が学士号を取得したカーネギー工科大学、現在のカーネギーメロン大学は、アンドリュー・カーネギーの寄付金で設立されました。カーネギー氏は当時、世界有数の大富豪でしたが、数々の図書館や教会オルガンを寄付したことでも知られるようになりました。彼は、必要な金額の半分だけを拠出する、いわゆるマッチングファンドを提供しました。残りの半分以上を地元からの寄付で賄われるようにすることで、「カーネギーからの贈り物」の価値を高めたのです。そのため、「カーネギー氏は、天に召された時に、光輪の金を半分だけ準備し、残りをマッチングファンドを当てにした」などと言う人もいます。

カーネギー氏のモットーは「我が心、仕事に込めたり(My Heart is in the Work.)」でした。始めに仕事ありきで、そこに魂を入れる、ということです。私もこの言葉は好きなのですが、私自身はどうかと言われれば、氏とはアプローチが逆になります。つまり、自分の好きなことをしていれば、その仕事が心を熱くしてくれると考えています。

私のこれまでの人生で最良のアドバイスをしてくれたのは、息子のディーンでした。ある有名大学の学長に就任しないかというお話をいただいた時のことです。息子は私に「お父さんはその仕事が楽しいと思えそうなの？」と聞いてきたのです。正直、間違っても楽しいと言えるような仕事ではありませんでした。息子は続けてこう言いました。「引き受けないほうがいいよ。お父さんは自分が楽しいと思えないことはうまくできない人だから」。私は彼の的を射た忠告を受け入れてその申し出をお断りしました。

私はこれまでずっと楽しんで仕事をしてきました。私にとって、仕事は遊びのようなものでした。もういつ引退してもおかしくない年齢になりましたが、私の心はまだ仕事と共にありますので、まだまだ仕事を続けていきたいと思っています。逆に言えば、私の心を熱くしてくれる仕事はまだあるということです。今の私の心を熱くしてくれている仕事については後ほど少し触れてみたいと思っています。

なぜ自分の仕事が好きなのか

私は生まれた時からエンジニアになることを運命づけられていました。母は学校の

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

先生で、子どもたちの教育に心血を注いでいました。私には、単語の音節を一つひとつはつきりと発音するようにと、発声法を教えてくださいました。私に詩を暗誦させ、意味がしっかり伝えられているか、韻律やリズムは適切であるかなどを確認してくれました。私は今でも、当時覚えた詩のいくつかを繰り返し暗誦しては楽しんでいます。こうした財産を私に残してくれた母に感謝します。

母は車の運転が嫌いでした。それでも、何週間も私が母にせがんだので、ブルックリン橋を見に連れて行ってくれたことがあります。その時、母と一緒にブルックリン橋を端から端まで嬉々として歩いて渡ったことを今でも覚えています。また、ニューヨーク市を走り抜けて、私と兄をベル電話研究所のクロード・シャノン博士に会いに連れて行ってくれたこともありました。そこで私たちは博士が作った迷路探索ネズミを見せてもらいました。当時は適当なトランジスタがまだなく、リレーによる論理回路が使われていました。それから10年ほど経った1961年、私はシャノン先生に博士論文の指導をお願いしました。先生は私の依頼を引き受けて下さっただけでなく、昔、二人の少年の訪問を受けたことを覚えていて下さいました。このクロード・エルウッド・シャノン先生は1985年の第1回京都賞基礎科学部門を受賞されていますが、私たち兄弟を先生のところへ連れて行ってくれた母に、改めて感謝したいと思います。

父は優秀なエンジニアでした。1920年代にロンドン大学で土木工学の博士号を取得しているのですが、学校教育で得た知識より遥かに重要で実用的な知識を豊富に身に付けていました。機械がどのように動き、なぜ動かないかというようなことを「直感的に」理解していました。私は、父がものを修理するのを見て実学的知識の大切さを学びました。また、父は、トイレのタンクが一杯になった時に中の浮きが水を止める仕組みや、蒸気機関車のシリンダーが車輪を駆動する原理などを教えてくださいました。電気の三相交流の仕組みについても、それを理解するだけの数学的知識が私にまだなかった頃に教えてくださいました。父は学問としての数学の知識に留まらず、それよりも遥かに大切なこと、数学は何のために必要かという命題について深く理解していました。単に方程式を解くだけでなく、どういった方程式を解き、その解は何を意味するのか、ということを知らなければならないのです。こうした理解力を授けてくれた父に感謝します。

父と母は、二人して息子たちに暗算を教えようとしていました。車で遠出をしている時には、よく家族で「数字当て」ゲームをしました。「私は頭の中に数字の一つ思い浮

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.

かべています。その数字を二乗して2を足すと18になります。その数字は何でしょう？」といったものです。両親はそうした問題を際限なく繰り返し、また、問いのレベルも私たちが成長するにつれて段々と難しくなっていました。おかげで私たち兄弟は悪戯をする時間もなく、学校で習う何年も前に代数を学ぶ機会を得たのです。

小さい頃、私にはお気に入りのおもちゃが二つありました。一つはティンカートイで、木製の棒や円盤状のパーツを使って動くおもちゃを作ることができました。棒の長さや円盤の穴の角度が色々あって、それらを組み合わせて三角形の構造単位を組み入れれば強度が増します。精密な図面セットが一式入っており、それを見れば複雑な構造物の組み立て方が分かりました。当時、私は6歳か7歳でしたが、図面どおりに組み立てるようにと、母にいつも言われていたのを覚えています。それから、母がパーツを買い足してくれる度に、私の作品は大きくなっていったのですが、そのティンカートイもいつの間にかどこにいったのか分からなくなってしまいました。

もう一つのお気に入りのおもちゃは、歴史あるドイツの建造物ブロックで、リヒターブロックと呼ばれるものでした。見た目は石でできているようなのですが、実際には、砂と細かい粉末を重合亜麻仁油で固めたものであることを後になって知りました。このブロックには詳しい図面が層ごとにあって、立派な建造物を作り上げる方法が示されていました。設計図の読み方を母に教わった私は、英語の文章を読むことができるようになる前にリヒターブロックの設計図を読みこなし、建造物を作ることができるようになっていました。このブロックと図面は今でも持っています。

私が通っていた学校では、それぞれの家庭で教科書に保護用の紙製カバーをつける決まりになっていました。友達は皆、ハーバード、イエールなどの有名大学から買ったカバーを使っていたが、母はそのような上等のカバーを買う余裕はないと言って、父が仕事で使っていた青焼き図面でブックカバーを作ってくれました。その青焼きは、父が建設したダムの内部構造を示す図面で、タービンの位置や送水方法などが示されていました。おかげで私は高校に入る前に本物の青焼き図面が読めるようになりました。

私は父に手伝ってもらって、ガントリークレーンを作りました。この写真(Fig. 1)は学校新聞に掲載されたものです。その1年後には、蒸気シャベルの模型を作りました。モリソン・ヌードセンという建設会社のヘルメットをかぶった私と一緒に写っているのがその模型です(Fig. 2)。先日、ポートランド近郊のスワン島にあるビガー社の造船所を見学する機会がありました。そこには、私が組み立てた模型に良く似た

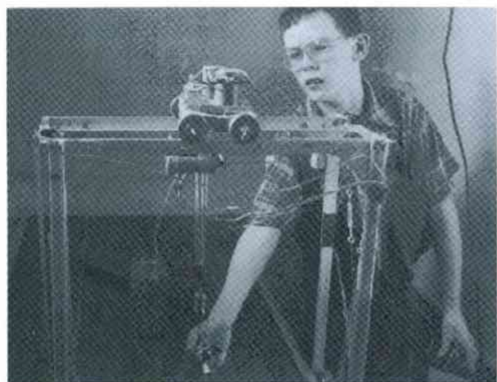


Fig. 1

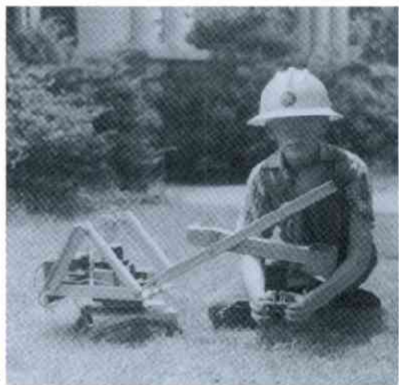


Fig. 2

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.

クレーンがありましたが、本物は600トンのものを持ち上げることができました。造船所では、ツアーの参加者にと新しいヘルメットをいただきました。そのヘルメットとクレーンが、こうした場所を父と訪ねた60年前の記憶を呼び覚ましてくれました。

幾何学的構造

今思えば、私は小さい時から図面と実物の両方を見る機会に恵まれていました。ティンカーToyであれ、リヒターブロックであれ、図面を見て実際の作品を生み出していました。また、父が設計したダムを図面の見方を学ぶ一方で、本物のダムをしばしば訪ねたりしていました。図面は実物とは異なります。実物の原型(モデル)にすぎません。

エンジニアリングは実物のこのようなモデルに依存しているのです。エンジニアリングモデルの代表的なものは図面ですが、縮尺模型や数学の方程式の形態をとることもあります。エンジニアリングで最も重要な部分はモデリングなのです。モデルが現実には忠実であれば、それは上手く機能し、飛行機が飛び、ビルが建ち、橋が交通量の重みに耐えることができるのです。逆にモデルが現実をうまく捉え切れていないと、事故が起こります。海洋の挙動モデルに欠陥があったために、日本は、先日のような大きな被害を受ける結果になったのです。

私の思考において、幾何学的構造は常に根本的な部分を占めてきました。モデルと実物は同じ幾何学的構造を共有しています。ティンカーToyの実物はその図面と同じ幾何学的構造を有しています。リヒターブロックもそうです。幾何学的構造が正しいから、物がぴたりと合うのです。機械工学における幾何学的構造の役割は明白です。しかし、私は電気技師なのです。電気工学における幾何学的構造の役割はどこにあるのでしょうか。

私は、電気回路の単純な幾何学的モデルを自分で開発しました。この単純な幾何学的モデルでは、力は電流を、位置または長さは電圧を、そしてバネは抵抗を意味していました。バネに引っ張る力を加えるとバネの長さが変化します。つまり電流を流すと抵抗にかかる電圧が変化します。バネを伸ばすためには、力を加えなければなりません。抵抗における電流と電圧の関係を示すオームの法則の公式は、バネにおける力と伸びの関係を示した等式に等しいのです。つまり、私の単純な幾何学的モデルは現実には忠実なものなのです。

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

私はトランジスタやコンデンサに関しても単純な幾何学的モデルを持っています。このモデルでは、天井から吊り下げられた鐘のひもを引く修道士がトランジスタ、自動車のショックアブソーバがコンデンサです。こうしたモデルの詳細はいささか専門的になりますので、説明は別の機会に譲ることにさせていただきます。ただ、トランジスタの動作やコンデンサの挙動に関する私の単純な幾何学的モデルも、現実に忠実であることだけは申し上げておきます。

私にとって自分で作った単純な幾何学的モデルは役に立つものです。「心の目」でこうしたモデルの動きが想像できるので、回路がどのように動作しているかを「見せて」くれるからです。こうしたモデルで回路を「感じる」ことができます。私の数学的知識を補う絵として見せてくれます。数学は正確な答えを教えてくださいが、幾何学的モデルはどういった数学的手法を用いるべきかを教えてください。

コンピュータ描画

カーネギー工科大学では機械製図が必修科目でした。理由は二つあります。一つは設計図の読み方を教えるためですが、私は、教科書カバーとして使った青焼き図面で、すでにそのスキルを習得していました。二つ目は、紙の上に鉛筆を使ってきれいな図面の描き方を教えるためでした。私は手先が器用でなく根気もあまりなくて、製図は好きではありませんでした。最悪だったのは、消しゴムを使うと製図用紙の上に見苦しい痕跡が残ったり、消しカスだらけになってしまったりすることでした。

その頃からすでに私はコンピュータが使えるらと思っていたのですが、当時のコンピュータと言えば、部屋一つ分程の大きさでしたし、使うにも大変な費用がかかりました。当時は「オンライン処理」でコンピュータを使用するといったことも始まっていませんでした。「バッチ処理」と呼ばれる方法が主流でした。ユーザーは、パンチしたカードをひとまとめにして、コンピュータにデータを与えるオペレーターに渡しました。そして、数時間以上も待ったのちに、ユーザーはすべて大文字で打ち出された大量の紙を受け取りました。

1960年、私は大学院生としてマサチューセッツ工科大学(MIT)に入学しましたが、そこで描画に使えるようなディスプレイを持ったコンピュータ数台に出会いました。コンピュータで製図ができるかもしれないとそれとなく思ったのはその時が初めてでした。このコンピュータで線をまっすぐに引けるだろうか。このコンピュータで

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

見苦しい跡を残さずに描いたものを消せるだろうか。残念ながら当時MITのキャンパスにあったコンピュータはこうしたタスクを実行するには小さすぎました。

ある年の夏休み、私はMITのキャンパスからは離れたところにあるMITリンカーン研究所でアルバイトをすることになったのですが、そこで出会ったのが当時世界最高のコンピュータ、TX-2でした。図を表示できるディスプレイやタッチペンとして使うことのできるライトペンが備わっていました。メモリーも巨大で、0.25メガバイトほどありました。また、速さもずば抜けていて、1秒間に8万回の演算をすることができました。今の基準では原始的なスペックですが、当時としては世界最高のコンピュータだったのです。リンカーン研究所は、数千個のトランジスタが巨大コンピュータでどのように動作するのかを確認するためにこのコンピュータを作ったのです。

TX-2を設計したのはウェス・クラークという人物です。当時一介の大学院生に過ぎなかった私にとって、彼は自分よりも随分と先輩に思え、少し怖い存在でしたが、その後私たちは親友となり、現在も親しくお付き合いさせてもらっています。ずっと後になって、彼は私に、TX-2は実は私のために作ったのだと教えてくれました。もちろん、作っている時には私がどんな人物であるかも彼は知らなかったはずですが、この話は本当のようです。スケッチパッドを動かすためにはTX-2のパワーがどうしても必要で、実際に使い切りました。当時のバッチ処理型コンピュータとは違い、TX-2は各ユーザーが何時間も同時に使うことができました。スケッチパッドの実験と改良を行うには、この程度のアクセスを確保することが必要でした。

当時、コンピュータを利用した描画で何をすべきか、誰も答えを持っていませんでした。鉛筆と紙より優れた描画ツールとなるのか。そもそもコンピュータを利用した描画は紙の上に絵を描くのと違うのか。もし違うとすればどう違うのか。当時の私は、そうしたことはお構いなしに、ただ画面上にきれいに絵を描くことだけを考えていました。そして試行錯誤を繰り返していきました。赤ん坊が四肢の筋肉を収縮させて腕や足の動かし方を学んでいくのと同じく、科学者も知的筋肉を柔軟に動かして物質の動作原理を学んでいく必要があるのです。このような試行錯誤のプロセスが大切な作業であると今は分かっています。これが「リサーチ(研究)」なのです。

ところで、先ほど私は「コンピュータを利用した描画はどうあるべきか」ではなく「何をすべきか」と申し上げたことにお気づきになられたでしょうか。スケッチパッドは、コンピュータを利用した描画が紙を使った描画よりもはるかに優れていることを示してくれましたが、それはただ単にそれがそこに存在するだけでなく、実際に仕

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.

事をしてくれるからです。つまり、コンピュータで描いたものは動かすことができますが、紙の上に描いたものは動かすことはできません。また、コンピュータで描いた部品は、お互いにくっつけて、そのままにしておくこともできますが、紙の上に描いた図は単なる汚れ(無価値なもの)でしかありません。紙の上に描いた図の持つ意味は人の心の中でしか存在しません。コンピュータを利用した描画は、固有の意味を持つことが可能なのです。私が「どうあるべきか」ではなく「何をすべきか」と申し上げたのはこのためです。

TX-2で画面上に線を描くのは大変な作業でした。ディスプレイ自体にできることは、小さな画面上に配置された約百万ある光のドットを点滅させることだけでした。これで線を描こうとすると、その線に沿ってドットを点滅させなければなりません。ディスプレイにメモリーはありませんので、画面上に線を表示し続けるためには、同じドットを何度も繰り返して点滅させなければなりません。線は物ではなく一つのプロセスとなるのです。言い換えれば名詞的な概念、つまり「そこに置かれたもの」ではなく、繰り返されるプロセス、つまり動詞的な概念となるのです。TX-2は線を「描く」のではなく、線を「行う」のです。しかもその行為を何度も何度も繰り返して行うのです。線を消すのは簡単で、プロセスを停止すれば線はなくなります。私は、コンピュータを利用した描画がどのように見えるのかということだけでなく、それがどのように動くのかを学びました。

私は一本のムービーを作りました。当時色々ところで紹介され、今もネット上で見るができます。その中で、描画が形を変える様が捉えられています。知って頂きたいことは、当時、コンピュータの画面を実際に見たことのある人はほんの一握りしかいませんでした。そのため、コンピュータを利用して描画するという発想は全く新しいもので、ましてや絵が動くなど言わずもがなでした。

現在、コンピュータを利用した描画は二つの用途に用いられています。一部は紙の上の描画同様、知識や楽しみを人々に提供することを目的として使われています。アートや娯楽を目的としたコンピュータを利用する描画は、今では至るところで見られますし、科学技術計算に基づくコンピュータ出力の図面も一般的に用いられています。こうした描画は私たちが理解できるようにしてくれますが、コンピュータ自身に対しては何も働きかけてはいません。こうした描画はコンピュータの出力であり、ものの見え方を我々に見せてくれるものです。

コンピュータを利用した描画のもう一つの用途は、コンピュータのために事実の定

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

義を行います。コンピュータ支援設計で使われる描画は、紙の上の描画とは異なります。私たちはコンピュータに、自分たちが描いた回路をシミュレーションさせたり、結合体の挙動を計算させたり、構造物の重さや強度を計算させたりします。こうした描画はコンピュータへの入力であり、コンピュータにももの見え方を教えるものです。

1975年、私はボブ・スプロールとボブ・シューマッハーと共著で *A Characterization of Ten Hidden Surface Algorithms* (10の隠面アルゴリズムの特性) という本を出版しましたが、この本がコンピュータグラフィックス分野における私の最後の研究成果をまとめたものとなりました。私たちは当時知られていた、立体のように見える絵を描くための10個のアルゴリズムすべてを分析し、そのすべてにソーティング(並べ替え)の工程が含まれていることを見つけました。コンピュータを使ってもっと上手に絵を描くためにはソーティングの手法を改善する必要があると認識しました。しかし、このソーティングの改善作業は私にはつまらないものに思えました。そう、私の心を熱くはしてくれなかったのです。

第二部： これからのこと

1976年、私はカリフォルニア工科大学でカーバー・ミード教授と知り合いになりました。当時、電子工学の世界で「革命」が進行中でした。ミード教授は集積回路(IC)チップの設計をシンプルに捉えていました。ICチップはトランジスタと配線でできた幾何学的パターンを使って論理演算を行います。私はその幾何学的構造を面白いと思いました。ICチップはトランジスタと配線で構成された電子回路ですが、私はその回路も面白いと思いました。また、ICチップは論理演算を行います。私はその論理回路にも関心を抱きました。この三つが合わさって私には抗しがたい、複雑ではあるが知的好奇心をそそる設計課題になりました。それまで出会ったことのなかった仕事が私の心に火をつけたのです。

ミード教授は、超LSI(VLSI)回路の分野における名著をリン・コンウェイ博士との共著で出版しています。『超LSIシステム入門』という本ですが、『ミードとコンウェイ』と言った方が通りがいいかもしれません。ミード教授と私はカリフォルニア工科大学の研究仲間でしたが、コンウェイ博士はゼロックス社のパロアルト研究所(PARC)で私の兄、パート・サザランドのチームに所属していました。二人はこの共著を兄に捧げてくれています。

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

ゼロックスPARCでは、マルチプロジェクト・チップが発明されています。簡単に言うと、小規模のプロジェクトを数多く集めることで膨大なチップ製造コストの負担を軽減しようというものです。それから間もなく、米国政府は、学生が設計したマルチプロジェクト・チップの製造を請け負うMOSISというサービスに資金を提供しました。おかげで一夜にして— 私にはそう思えました—、およそ100ものIC設計に関するコースが設立されました。その結果、この技術分野における知識が開花し、IC技術の爆発的な進展を支えていくのに必要とされるエンジニアが業界に送り込まれました。

私はこれまで二つの大きなパラダイムシフトに関わってきました。一つはスケッチパッドに関するもので、コンピュータを使って絵を描くという行為を一変させました。私自身はそのような変化を起こそうなどと考えたことはありませんでした。ただ面白いと思って行った結果がそうだっただけなのです。京都賞を頂戴し本当に光栄に思いますが、決してそうした栄誉を求めて研究を続けてきたものではありません。

二つ目のパラダイムシフトはICに関わるものです。私はなにもIC設計に秀でた技術者を増やしてやろうと考えたものではありません。中にはそう考えた人もおられたかもしれません。例えば、リン・コンウェイ博士は、その重要性を十分に理解してICコースを担当する教員の指導にあたっていたと思います。私はというと、日々、技術的に一番興味をそそられることをやっていただけでした。

次に私は、三つ目の重要なパラダイムシフトを目撃したいと思っています。私にとって、これはこれまで追い求めてきたものとは大きく異なる目標となります。これまではパラダイムシフトを自ら引き起こそうと思ったことなどありませんでした。後の用途の拡がりなどは考えず、ひたすらより良い技術を求めて努力してきました。パラダイムシフトとは、より良い技術の結果として、自然にもたらされたものだったのです。

自己タイミング

1988年に出版した『マイクロバイブライン』という著書で、私は自己タイミング型パーツでパイプラインを形成する方法を示しました。アイデアは至ってシンプルです。デジタルシステムの各パーツにそれぞれのペースで仕事をさせるのです。自己タイミングという考え方は革新的なアイデアであると言えます。というのも、現在ある

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.

デジタルシステムは、そのほとんどすべてが「クロック」と呼ばれる外部リズムにタイミングを依存しているからです。

今日のデジタルシステムは、そのほとんどすべてが周期性を持つ外部クロックのビートに合わせて時間を短い「クロック周期」に量子化しています。どの論理ステップがどのクロック周期に起こるかを設計者は正確に把握することができるため、リズムを刻むクロックは重宝されています。また、次のステップに進む前に必要とされる事象がそれまでのステップで既に起こったかどうかをチェックすることもできます。つまり、クロック周期は論理設計の仕事を簡単にしてくれるのです。

一方、クロック周期はチップの電気設計を複雑にします。一つひとつの論理演算は、選ばれたクロック周期に合致するものでなくてはなりません。つまり、速度の速いシンプルな論理回路は待たないといけませんし、難しくて時間の掛かる論理タスクは急がなければなりません。一様な遅延は、設計コストが高むだけでなく、エネルギーの無駄にもなります。最も厄介なのは、信頼できる形でクロックを提供する設計自体に力を注がねばならず、電気エネルギーも浪費されることです。

パラダイムの変更

クロックを持つ回路設計のパラダイムは、現在、広く利用されていますが、私には、実用面でも原理面でも欠点があり、全く行き詰っているように見えます。実用面での欠点とは、クロック同期型パラダイムは論理設計、電気設計、幾何学的構造設計という三つのハードの設計に関する問題が絡んでいることです。チップの幾何学的構造や論理回路が変われば、配線の数、長さ、位置も変わり、電気設計に影響を与えます。また、チップの論理設計や電気設計が変われば、その部品のサイズが変わり、幾何学的構造設計に影響を与えます。さらに、電気または幾何学的構造設計が変われば、クロック周期を超えて設計の妨げになるかもしれず、論理回路に手を加えなくてはなりません。このように、クロック同期型パラダイムには三つのハード設計作業が絡むため、大規模なクロック同期型チップの設計には非常に大きなコストが掛かります。こうした複雑な状況は、各信号を一定のクロック周期に合致させる「タイミング収束」と呼ばれるプロセスにおいて最も顕著に見られます。

クロック同期型設計に内在する根本的な欠点を説明するために、喩え話を一つさせていただきます。先日、NASAの火星探査機、キュリオシティが火星に着陸し、多く

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.

の人々がその光景を目撃しましたが、火星は非常に遠いところにあり、地球に光が届くまでに13分も掛かります。ですから、地球からはキュリオシティが着陸するのを見るのは不可能で、私たちが見たものはキュリオシティが13分前に無事に火星に着陸したというレポートに過ぎないのです。つまり、地球からキュリオシティの着陸を操作するということは不可能なのです。13分の通信遅延があるので、キュリオシティは自己タイミング型でなくてはならないのです。

ICは、新たに開発されるたびに、トランジスタがより小さく、配線がより細くなっています。トランジスタが小さくなればその動作も早くなりますが、配線が細くなるとその信号伝達スピードは同じか、遅くなってしまいます。もちろん、光の速度とは比べ物にならないほどゆっくりとしたスピードです。トランジスタを小型化すると、クロック信号の、より精度の高い「同時」分配が必要になりますが、配線が細くなると「同時」は困難さを増します。現在、クロック信号の「同時」分配は、最新のチップでは限られた部分で実行されているだけです。もはやクロック同期型パラダイムをチップ全体に適用することはできないのです。

私は自己タイミング型のパラダイムが望ましいと思っています。自己タイミング型なら、独立モジュールは実際に独立しているので、モジュール性を確保することができます。独立モジュールは共通のクロックを必要とせず、それぞれベストの速度で処理することができます。また、自己タイミング型なら三つの設計課題を解決することも可能です。各パーツが自分に必要なデータを待つので、論理回路の正確性は、幾何学的構造に依存しないだけでなく、電気設計への依存度も随分低くなります。配線を長くしたり短くしたりして幾何学的構造を変えても論理回路は正確なままで、タスク処理に掛かる時間が変わるだけです。同様に、電気的変化や製造ばらつきがあっても論理回路の正確さは変わらず、タスクに要する時間だけが変わります。最後に、自己タイミング型では通信遅延によって発生する時間と空間の基本的関係が認識されるので、自己タイミング型システムでは自由にサイズを変えることができます。

現在はまだ稀にしか使われていませんが、私は、デジタルシステムの設計において、自己タイミング型が有用な代替技術になって欲しいと思っています。また、自己タイミング型を利用することによって大きな恩恵がもたらされるものと信じています。そのため、私はデジタル設計のパラダイムを変えるべく、努力を続けています。

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.

何をどうすれば？

このパラダイムシフトを目指し、現在、私はこれまで慣れ親しんできたものとは全く異なる課題に挑んでいます。私の思考が前向きな時は、「成功の物差しは？」、「変化を加速させる方法は？」、「パラダイムシフトを促すレバレッジポイントというものは存在するのか？」、「私には既に明白だと思われることを周りにも分かって欲しいが、どうすれば簡単に説明できるのか？」といった問いが頭の中を巡ります。

逆に、後ろ向きな考えが頭の中を支配している時は、「クロック同期型パラダイムはすっかり定着しているのでそれを変えることは不可能では？」、「そもそも私の考えるパラダイムシフトは可能なのか？」、「私の目標は非現実的なものでは？」「目標の合理性を検証する方法は？」などと自問しています。

しかし本当の問題は、私の心を熱くしてくれるものを見つけ出せるかということにあります。自分は本当にパラダイムシフトを起こしたいのか。変化を起こすこと自体を目標とすべきなのか。あるいはこれまで通り、パラダイムシフトは人に任せて自分は好きな仕事に専念すべきか。

ここでこのように色々と申し上げたのには二つ理由があります。一つ目の理由は、残された人生で何をすべきかというジレンマと折り合いをつける必要があるからです。現在自分が抱えているジレンマについて書いたり、話をしたりすることによって解決の糸口が見つかることを願っています。二つ目の理由として、京都賞の受賞者でさえ自らの目的に対してジレンマを感じている、少なくとも一人の受賞者がそうしたジレンマに陥っているのだということを皆さんに知ってほしいということです。その意味で、今回、「心を熱くするもの」を探し求める機会を私に与えて下さった稲盛財団の皆様に感謝いたします。

将来のイメージ

私が抱えているジレンマに関して、大事なことが三つあります。まず、レバレッジポイントがあるかもしれないということ。つまり、幅広い用途に対応できる、柔軟な自己タイミング型ICチップが今後開発される可能性があるということです。例えば、私が小さい頃遊んだおもちゃのブロックのように、ユーザーにたくさんの論理パーツが提供され、それを積み上げて論理構造を組み立てていくといったようなことで

す。私の組み立てたブロックが重力によって一定の位置に固定されていたように、そうしたチップはユーザーが作り上げた論理結合を覚えているのです。実は、そうしたデバイスは既に存在しています。フィールドプログラマブルゲートアレイ(FPGA)です。

クロック同期型FPGAは様々なところで使われていますが、今あるクロック同期型設計では、ユーザーはクロック同期型システムしか作ることができません。クロック同期型FPGAは、クロック同期型設計に関する新しいアイデアの検証用として一般的に使われています。今後開発されるであろう自己タイミング型FPGAは、ユーザーが自己タイミング型の設計に関する新たなアイデアを検証することを可能にしてくれるでしょう。

ひょっとすると自己タイミング型FPGAがレバレッジポイントになるかもしれません。何故なら、それによって自己タイミング型回路の普及が促される可能性があるからです。例えば、自己タイミング型に関する何千もの実験のベースとして使われたり、多数の学生の教育に活用されたり、自己タイミング型に関する新たな用途の基礎となる、といったことが考えられます。自己タイミング型FPGAチップは、自己タイミング型のパラダイムが花開く、肥沃な土壌となる可能性を秘めているのです。

二つ目の大事な点ですが、自己タイミング型FPGAチップは作っていて面白いだろうと思うのです。何故かという、それによって提示される設計課題は、私にとってはかなり大変なことです。私が生きている間にはやり遂げることができそうだからです。最も大事なのは、私の専門分野における最高の研究者と一緒に仕事をする機会が得られるということです。このプロジェクトは、素晴らしい同僚と肩を並べ、誇りを持って仕事をするチャンスを私に与えてくれるのです。

三つ目の大事な点は、他人に変化を強いることはできないということです。私の気に入ったアイデアであってもそれを他人に押し付けることはできません。無理やりパラダイムシフトを起こすことはできないのです。しかし、私にはそのお膳立てをすることはできます。最も期待できるのは、自己タイミング型に簡単にアクセスできるようにして、周りがその価値に気付いてくれることです。パラダイムシフトが起こるのか、起こらないのか、それは時間がたってみなければ分かりません。

私が為すべきことは、これまでと変わらず、専門的な研究に注力することです。私の心を熱くしてくれるのはそうした仕事なのですから。パラダイムシフトが起こるのか否かに一喜一憂してはいけないうるのです。レバレッジポイントが存在するという希望があり、やりがいのある技術的チャレンジがあり、有能な仲間がいるとい

うことで十分ではないかと思うのです。結果としてパラダイムシフトが起こったのかという判断は歴史がすることです。「我が心、仕事に込めたり」であり、この仕事はまさに私の心を熱くしてくれるのですから。

稲盛財団 2012——第28回京都賞と助成金

発行 2013年9月30日

制作 公益財団法人 稲盛財団

〒600-8411 京都市下京区烏丸通四条下ル水銀屋町620番地

Phone: 075-353-7272 Fax: 075-353-7270

E-mail admin@inamori-f.or.jp URL <http://www.inamori-f.or.jp/>

ISBN4-900663-28-2 C0000