Special Session

Retrospectives: The Early Years in Computer Graphics at MIT, Lincoln Lab and Harvard

Chair: Jan Hurst, *EJH Associates*

Speakers: Michael S. Mahoney, *Princeton University*
Norman H. Taylor, *Androx, Inc.*
Douglas T. Ross, *SofTech, Inc.*
Robert M. Fano, *MIT*

I am Jan Hurst; welcome to Retrospectives. In early 1988, SIGGRAPH funded a project which was called Milestones, the History of Computer Graphics. We believed it was important to capture the early history since the graphics of tomorrow continues to build upon its past. We established goals, and one result was to begin a series of retrospectives which focus on specific aspects of the industry. Boston seemed like the perfect opportunity to focus on MIT, Lincoln Lab, and Harvard.

I hope that you will get a sense from the pioneers as to what graphics was like in its infancy in Boston. I hope that you will enjoy hearing about their experiences as I have. They have been absolutely charming and inspirational to work with, and I'd like to thank all of them at this time.

The speakers have many things to share with you, so I will be brief. I want to thank SIGGRAPH for supporting and encouraging this project and I want to thank the Pioneers for their willingness to share their experiences and their knowledge with all of us.

I want to introduce you to Michael S. Mahoney, who will act as the moderator for the retrospectives. Mike is a professor at Princeton, he is in the program in the History of Science, he is editor of the ACM History Series, and he is a distinguished member of the Milestone Advisory Committee. Mike.

Michael S. Mahoney
Princeton University

Thank you very much, Jan. As I said to the group when we had lunch today, I feel a bit ill at ease when I'm with them because they started the party about 40 years ago and I just walked in. Now I'm supposed to act as moderator of their activities and I'm supposed to introduce them to an audience that I suspect already knows many of them.

I'm going to keep my own remarks brief also because the story we are going to hear today is a story that really only they can tell. However, I would like to say a little bit about the value of panels of this sort.

Back in the mid '70s at a conference held, I believe it was at Los Alamos, on the history of computing, Dick Hamming of Bell Labs titled his keynote address, *We Would Know What They Thought When They Did It*, meaning that he was hoping that people doing the history of computing would get back beyond firsts to try to understand the mindset of the people who were doing the work and making the achievements that were being documented.

I didn't fully comprehend how deep a thought that was on his part until a colleague of mine told me a story about Jean Piaget, which may be apocryphal, but if it is, I'm going to tell it anyway, because it's too good to let go.

The story goes that Piaget was out one evening with a group of 11-year-olds and pointed out the moon to them and said, isn't it interesting that when we come out here each evening the moon is just a little bit higher in the sky? What do you suppose causes that? And the children said we don't know, professed their ignorance of it. And in the way that was his own, his highly manipulative, Socratic dialogue, he managed to get them to work out that the moon revolves around the earth about once a month and that this would account for the changing angle each evening.

About a month later Piaget was out with the same group of children, pointed out the same phenomenon and asked the same question and one 11-year-old popped up and said well, that's very simple. You see, the moon revolves around the earth about once a month and that accounts for the angle. And Piaget looked at the child and said — that's amazing, how did you know that? Oh, said the boy, we've always known that.

What we're trying to do with perspectives here is to get back to a time when people didn't know what they know now, try to understand what they were thinking before the questions became as clear as they seem to us now, and certainly before the answers did. And it's with that goal that we start today's proceedings.

Our first speaker is Norm Taylor who received his degrees from Bates College and MIT, and went on after the war to participate in the design and development of the very early Whirlwind Computer. He worked at Lincoln Lab where he was associate head of the computer division and then led the engineering effort for the four later versions, including the original, coincident, current magnetic memory on the SAGE FSQ-7. At Lincoln Lab he was a senior member of the SAGE Air Defense effort and was technical assistant to Dr. Jerry Weisner on the White House Gaither Panel.

Later on he joined ITEK as vice president of engineering and led the effort in the development of the Digigraphics system, later acquired by Control Data, where he became technical assistant to the president.

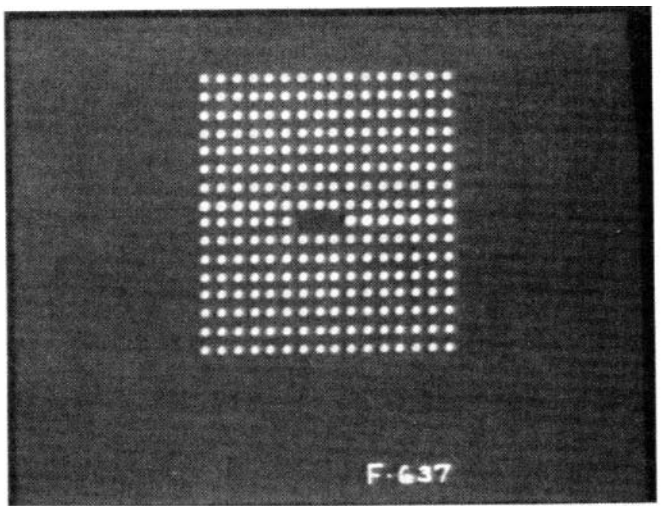
For the past 20 years he's been a consultant — both at Arthur D. Little and at his own firm, Corporate Tech Planning. And he told me just before we started that he is now adjusting to retirement, trying to stay active.

He received the first IEEE award for his work and publications on electronic reliability and was program chairman and later general chairman of the first and second national computer conferences, and Norm is going to talk to us this afternoon on the earliest displays on Whirlwind and SAGE. Norm Taylor.

Norman H. Taylor
Androx, Inc.

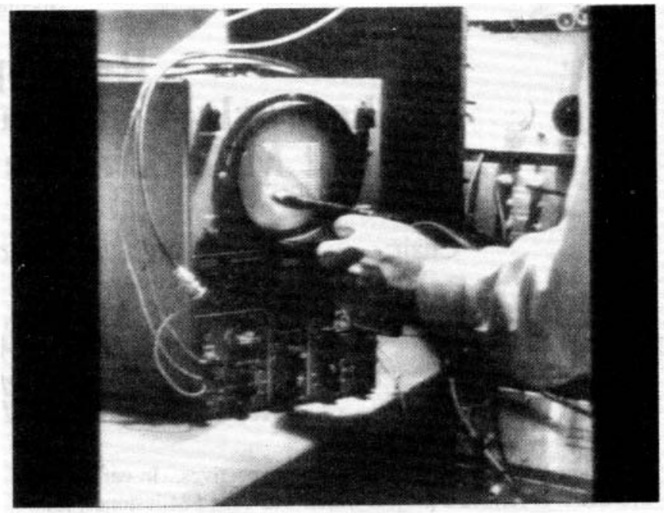
Mr. Chairman, Mr. Moderator, ladies and gentlemen, thanks for that generous introduction. When I was first called by Jan Hurst, she gave me an assignment to review the first 10 years of displays from '47 to '57 — and do it in 20 minutes. As I hung up the phone, the television showed President Bush talking about 1,000 points of light. All I could think of was, with due respect President Bush, we had 1,024 points of light in 1949.

I went to the Library and found we started in '48 with 256 points of light and here they are on slide one. As you will see we soon increased this to well over 1024.



— TAYLOR - SLIDE 1 —

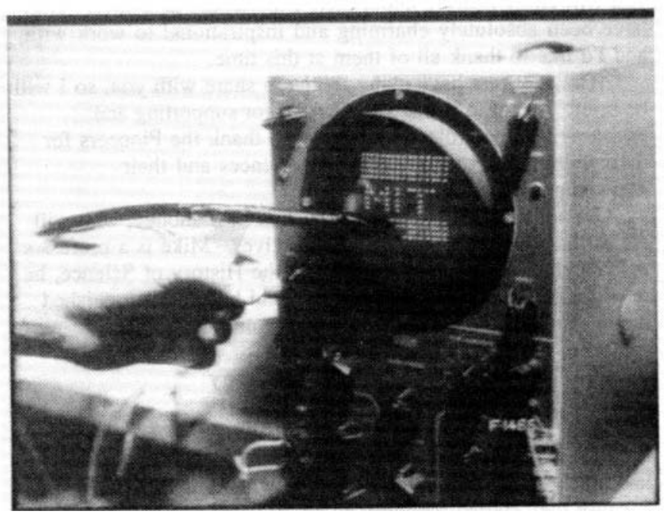
Now, these points of light really were a picture of the deflection system of the storage tubes in Whirlwind I. We had a program called the Waves of One. Waves of One ran through the storage tubes as a test. If we read a one, the program continued, and if it didn't, it stopped. We were asking how we can identify the address of that spot. So Bob Everett, our technical director, said "we can do that easily". All we need is a light gun to put over the spot that stops and we'll get a readout as to which one it is. So he invented the light gun that afternoon and the next day we achieved man machine interactive control of the display — I believe for the first time. This was late '48 or early '49. The next slide is a picture of one of those very early light guns.



— TAYLOR - SLIDE 2 —

Here it is. Later on Jack Gilmore is going to show you the original one, which is more like a light cannon than a light gun.

As soon as we displayed those spots on the tube, the public relations department became interested and Slide 3 is the early result.

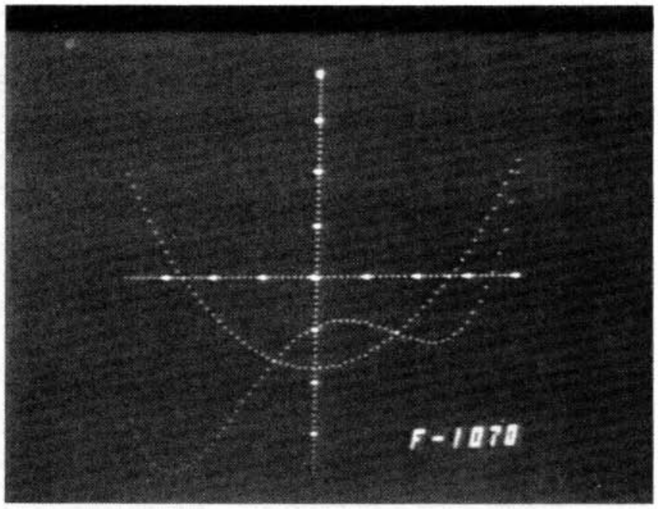


— TAYLOR - SLIDE 3 —

By erasing selected spots the MIT logo was produced.

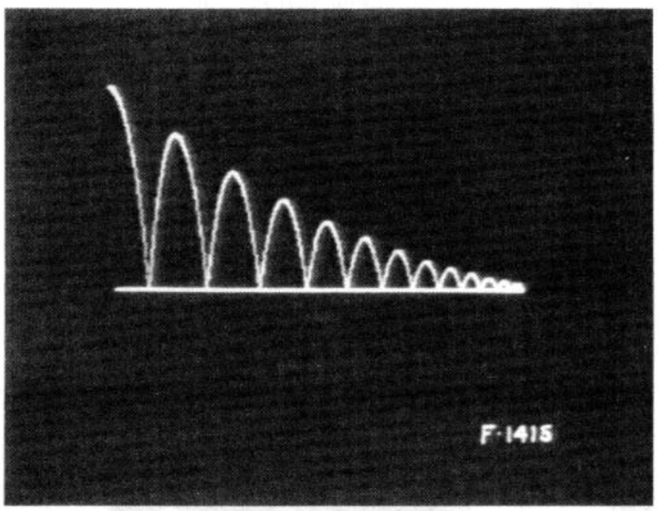
Whirlwind was a very popular machine. The dates on the back of these slides are 1947 and 1948.

In 1949, or '48, Edwin R. Murrow, who was the newscaster of the day, came to see Whirlwind, so we gave him a little welcome. This you see is just erasing selected spots of light to produce his name. It was clear that displays attracted potential users - computer code did not.



— TAYLOR - SLIDE 5 —

Then the mathematicians decided they could move a couple of curves that were displayed on a little test equipment tube.

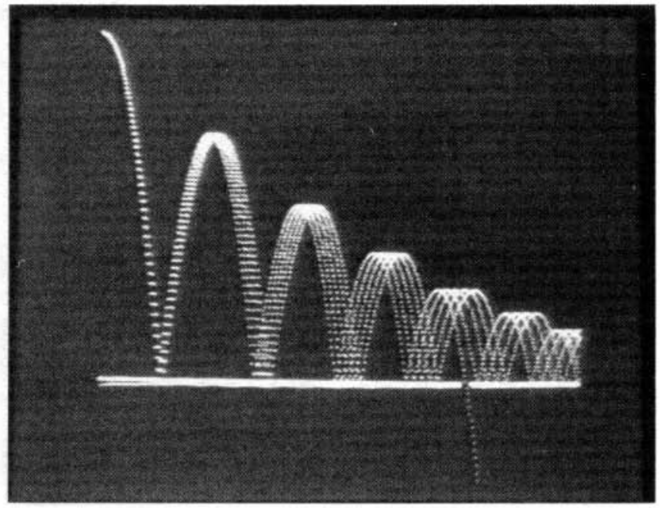


— TAYLOR - SLIDE 6 —

Charlie Adams, the original programmer, decided that we'd better go beyond static curves. And he invented what we call the Bouncing Ball Program, the solution of three differential equations.

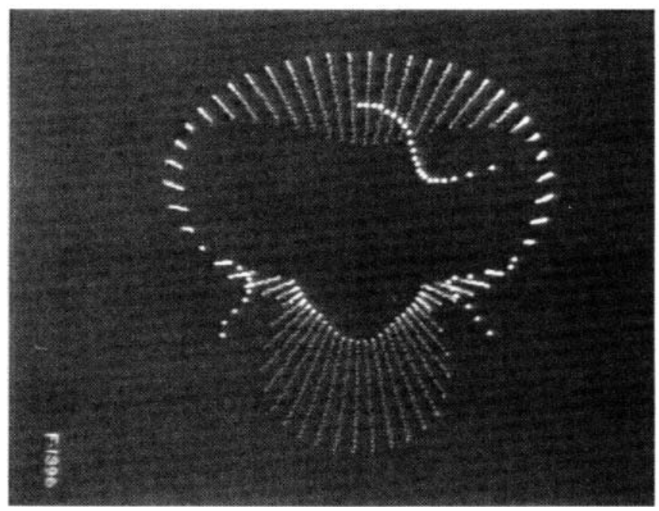
Now Whirlwind was a cantankerous machine. The storage tubes sometimes worked and sometimes didn't. In order to do the testing, we had built 32 registers of test storage. Twenty-seven of those registers were read- only memory; (we didn't have that word then. We just called it toggle switch memory). Five registers were flip flop memories.

The challenge was to put a program in those 32 registers that would run something interesting when the rest of the machine was down. Adams succeeded in doing this with some very clever programming. Jack Gilmore has a copy of that program. It might be the first display program ever written. This was in 1949.



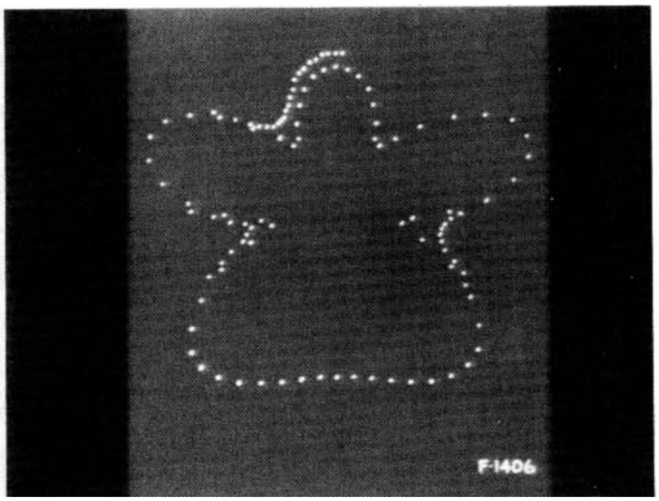
— TAYLOR - SLIDE 7 —

A little later Adams and Gilmore decided to make the first computer game, and this was also in '49. This is a more interesting display. You see that the bouncing ball finds a hole in the floor and the trick was to set the frequency such that you hit the hole in the floor. This kept a lot of people interested for quite a while and it was clear that man-machine interaction was here to stay. Anyone could turn the frequency-knobs.



— TAYLOR - SLIDE 8 —

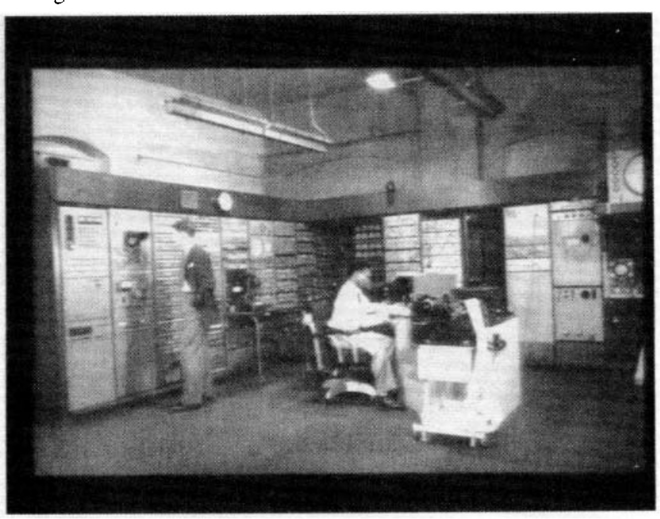
Shortly thereafter a masters thesis student named Dom Combelec decided to use a computer for design. His particular interest was the placement of antennas to make any array of patterns. Then he could change the array to find the pattern he wanted. This indeed was (slide 8) the picture of his search for the solution and the next one is the (slide 9) solution that he was looking for. (Perhaps this was the first computer aided design program.) So it was clear in 1950 that interactive displays were the real tool to link people with computers, but it was difficult to do and expensive to run fast enough to achieve response time.



— TAYLOR - SLIDE 9 —

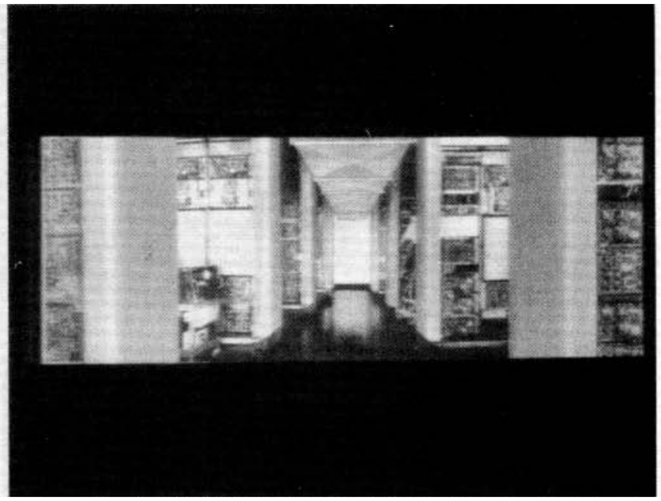
I might say that at this point we had gone above 256 points of light; we were up to 4,096 points of light, and this was just the beginning, as you all know.

This was really the end of the first phase in 1950. These were our simple displays. But I haven't mentioned what it took to make those displays and I'd like to give you just a little background on that. As mentioned above, it took a lot.



— TAYLOR - SLIDE 10 —

The next slide (Slide 10) is the room that we used as our display room. Actually it was a control room for the Whirlwind computer, and as you see, it's full of test equipment and the little scope that Jack Gilmore is looking at — this was 40 years ago — is a 5" Tektronix scope. That's all we had for displays in 1950. Keep in mind we were not trying to build a display here; we were building a computer. All we used the display for was testing the various parts of the system so displays were ancillary completely to the main event. Slide 11 gives you a glance at Whirlwind.



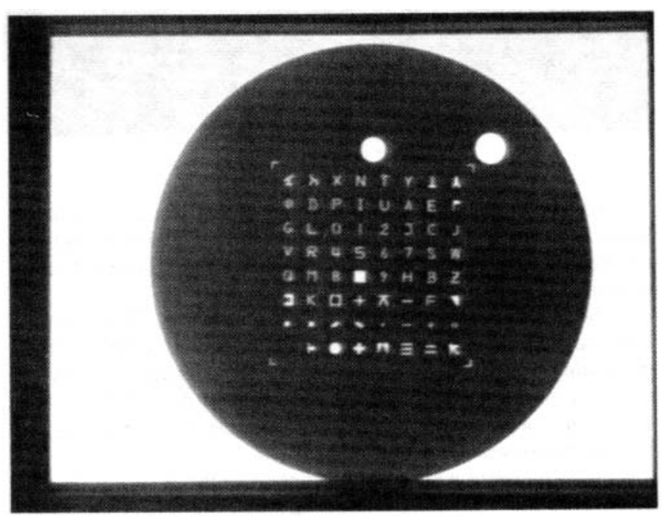
— TAYLOR - SLIDE 11 —

The computer itself took about a quarter of acre of electronics and had about 5,000 tubes in it. This was the problem we were solving for the first five years.

Shortly thereafter we moved to the second phase, when we were asked to make an Air Defense computer. We were tracking aircraft from a radar set for the FAA as several people at MIT mentioned it to the Air Force as a possible contender for the Air Defense program. (The Air Defense Command contracted with MIT to build a prototype of a system which became SAGE in the early 1950s

As soon as we received that contract, we needed more computers and more displays.

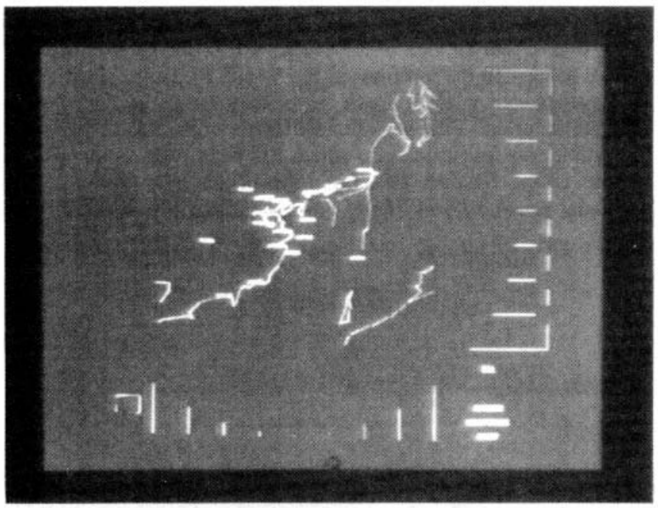
We realized that we needed a man-machine interactive display system to exercise control and soon found we needed a lot of displays to control over 400 aircraft simultaneously. My estimate of 32 consoles grew to 64 and then to 82.



— TAYLOR - SLIDE 12 —

So we searched for a tube that was a little more sophisticated than a simple point of light on a cathode-ray tube. We found that the Convair Company in San Diego had invented a Charactron. The Charactron was an unusual tube in that there was a character mask in front of the electron beam and as you defocused the beam slightly, you could run it through this mask, and it would produce one of these characters right on the screen. This wasn't as easy as it sounded because the beam was somewhat diffused, so the electron optics people had to redesign the optics to bring the focal point back a second time to a point in order to keep the integrity of those characters in any position. This gave us the tool to identify what these points of light meant and do it at high speed on 82 consoles.

The next slide (13) shows us what we actually did with this Charactron showing the New England coastline. Each one of those (Slide 12) shows the character matrix points are little letters which are very hard to read, but they actually defined locations of the key radars in the New England area. We used them not only for that, of course, but to determine whether the particular aircraft was a friend or a foe, a commercial or private plane. We had to determine the velocity of each plane in order to identify where you would plan an interception, and which interceptor should be assigned to it.

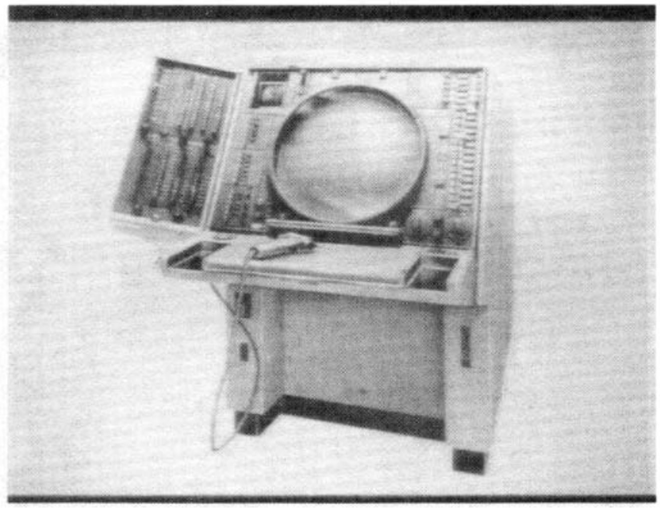


— TAYLOR - SLIDE 13 —

Along with the Charactron was another tube called a Typotron. The Typotron was the same idea, but it was a little storage tube. The storage tube was used to receive messages to and from various people (operators) that were going to control the action— for instance I assign a new aircraft to you, Mr. Controller. We think he's a bomber. You track him and identify his speed. If he agreed, we would pass the track to a weapons assigner. There was a lot of interaction between these people. (The Typotron became a message center among some 82 operators.)

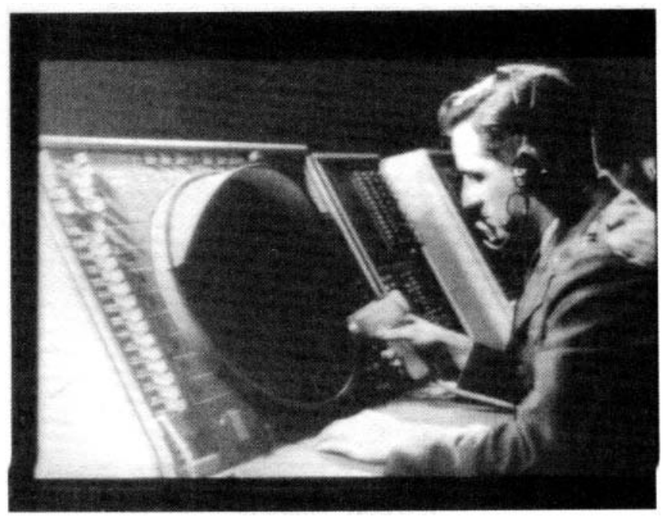
All along we used this light gun to tell the machine we were interested in a particular point of light. After selection, the operator sent instruction to the computer to tag each return.

So we assigned tags on each track. In order to do that we had to look at each one, track it for awhile and identify it as an aircraft. If it was a commercial airplane we usually could identify it because they all flew at the same speed and we had a flight plan in the computer. If it was an unknown it was tagged as a potential bomber. Unfortunately, the computer got busier and busier as we added more of these functions. Someone said later "if we knew how many of these functions we had to do, we wouldn't have started at all".



— TAYLOR - SLIDE 15 —

This is the finished console (slide 15). We turned the prototype design over to IBM, and IBM actually built the SAGE computer (FSQ-7s) as well as the consoles. For those of you who are going to the Computer Museum, there's a good, big piece of the FSQ-7 on display. I saw it a few months ago and I couldn't believe how big it was. It covered three floors of a full sized building in full operation until late 1978.



— TAYLOR - SLIDE 16 —

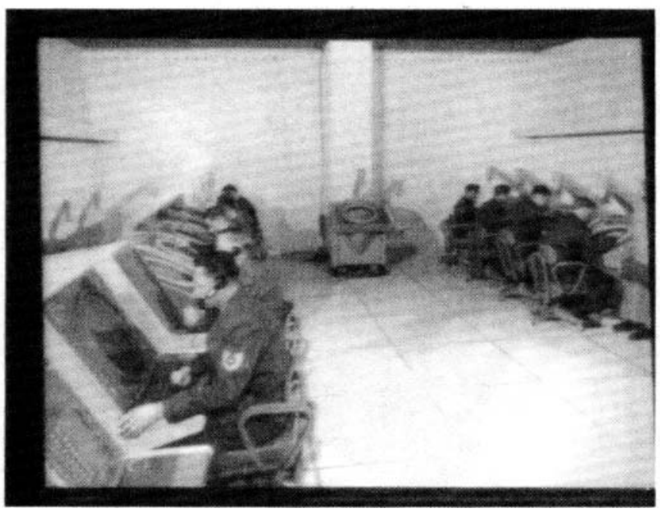
Slide 16 is an Air Force officer using the light gun, which got a little bit better looking as the years went by, and we're now up to 1955. In the left-hand corner you can see that little Typotron providing messages.

The reason the Typotron was to avoid distribution systems. We could send the information over one distribution system to either one of these tubes.

The distribution system was one of the biggest problems. In order to run a sector — New England was one sector — we had to provide 82 of these displays. You can imagine running 82 displays from one computer in 1956.

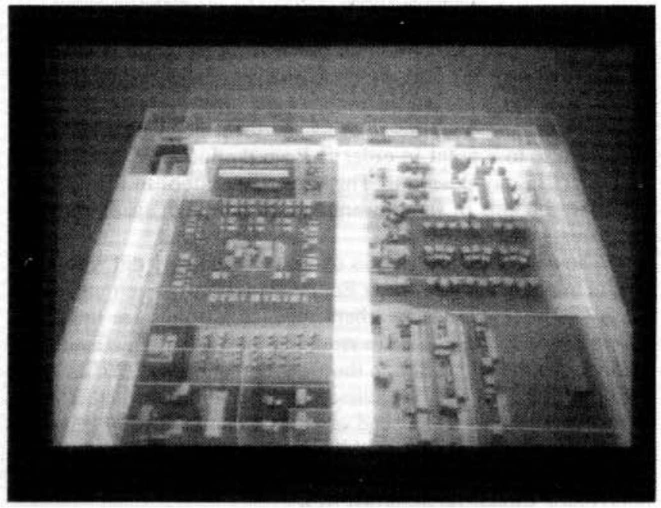
— TAYLOR - SLIDE 17 —

Slide 17 is the Cape Cod Display room. Before we built the SAGE, the first SAGE center, we put in what we called the Cape Cod network, which was a much smaller version of a SAGE network. It was running right out of the Barta Building at MIT and was running on Whirlwind. These consoles were of the prototype variety.



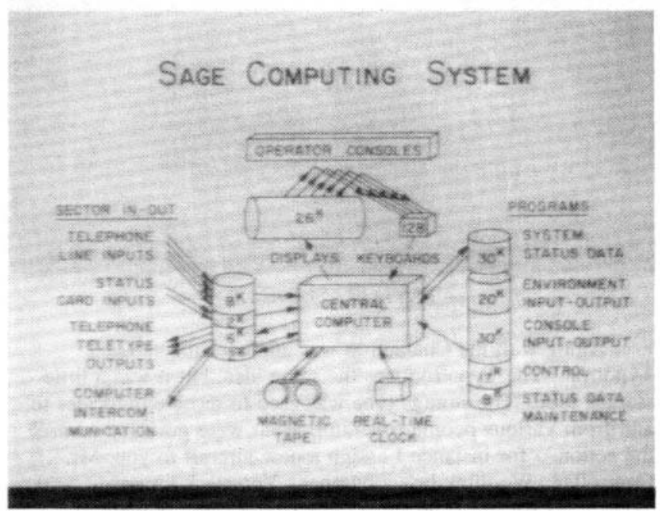
— TAYLOR - SLIDE 18 —

Now we come into final operation and here we find on Slide 18 Air Force people. These were enlisted men who operated and we had 82 of them in the first direction center.



— TAYLOR - SLIDE 19 —

To give you an idea of the scope of this display system, picture (Slide 19) the top floor of the SAGE Building where the 82 consoles were placed. Each one of these rooms provided a separate function. First was an acquisition of a track. Then the big problem of identification, which has been with us — still with us today. Second, the assignment of this track to a particular operator, and finally the assignment of weapons.



— TAYLOR - SLIDE 20 —

Perhaps from a technical viewpoint it might be well to let you know how hard it was to feed 82 scopes from one system. Slide 20 gives a diagram. By this time we had a display buffer drum system, some 26,000 words. Those were 32-bit words running the operator consoles. One high-speed bus (32 wires), ran all around the building. The display in question picked up the code and selected the display assigned to it.

Since the radars rotated each 15 seconds, the display cycle was repeated at this slow rate. So the tubes flashed a little bit. You can imagine how hard it was to orient your eyes to see new information every 15 seconds.

Since the new set of data came into it every 15 seconds, the computer had to refresh everything on that cycle. These other drums caught the data as they came in from the radars. So we were interlacing everything with 15 second intervals. But details of the FSQ-7 will take longer than my allotted 20 minutes and we will end here with the first ten years of computer displays. Thank you.

Moderator
Michael S. Mahoney
Princeton University

I forgot to explain the ground rules we had established. Each speaker is going to speak for about 20 minutes and then I thought what we would do is have two or three minutes for any questions from the audience specific to that talk, and that will leave us some time at the end of the session for more general questions across the speakers or from commentary from the floor, as difficult as it is to see.

Are there any questions that anyone would like to address to Norm Taylor? Good. Thanks for the — okay, here we go.

Q. You mentioned that the Charactron tube — then we saw the display there where the characters — and there were also vectors showing the outline of the New England coast. How did you generate those?

TAYLOR: I don't remember exactly. I think there were some vectors on that matrix and I think we added them together then. But we didn't have individual vector generators in this time period. So that was the reason we went to the Charactron. It was very hard with 82 different consoles to have individual anything. So we did everything centrally and I think the reason the coastline was so crude is that we used lines — if you want to actually go back to that slide, I don't know which one it was. It gives us that.

ROSS: The Whirlwind analog scopes and the Charactron both had analog vector generation, sweeping X & Y deflections.

TAYLOR: Okay, we did have an analog sweep, and I don't know how we transmitted that to the consoles. When I thought about that I tried to remember and I couldn't remember. Remember the 82 consoles, Doug? That didn't make it easy.

Q. Thank you. This work is so exciting.

Q. This Charactron tube — is this the same one that went in to the Stromberg Carlson the 4020 film recorder later?

TAYLOR: I believe so. I believe Stromberg Carlson was the source of the Charactron. The inventor was a fellow named Joe McNarny, I failed to mention that. It was I think Stromberg either sold it to Convair or was in a joint venture. As I remember, the Stromberg name was involved and I think it was the only one around at that time.

The Typotron tube which I mentioned was made by Hughes Laboratory. We went there. They did have a storage tube and we asked them could they put a Charactron on a storage tube mask and they were delighted to do that.

We had a little trouble funding the Typotron because Hughes always wanted a development contract, and you know about how hard that is. So we just plain ordered the 50 of them at a fixed price before they were designed and somehow it all worked out.

Moderator
Michael S. Mahoney
Princeton University

Thanks very much, Norm. Our next speaker is Doug Ross, Douglas T. Ross, who began his career as a graduate student in math at MIT where he was quickly lured into computing, serving as head of the Computer Applications Group from 1952 to 1969, when he left MIT to form SofTech, Inc., which he served as president until 1975 and since then as chairman.

He's made seminal contributions to several areas of computing, ranging from automatic programming of numerically controlled tools, the computer language APT, through Computer Aided Design, his Automated Engineering Design System, down to software engineering, the method of structured analysis design technique, which he puts generally under the heading of man machine collaboration.

He's an old hand at historical gatherings, having reported on APT at the conference on the History of Programming Languages held in 1978, and more recently on the early development in the History of Personal Workstations in 1986.

Today he is going to talk about working computer graphics from Whirlwind through Sketchpad and beyond with a little bit about APT. Doug Ross.

Douglas T. Ross
SofTech, Inc.

Thank you very much, Mike. Actually, I'm going to be trying to cover a 17-year span from 1952 when I started on Whirlwind — I'm just a youngster compared to Norm, you see — through the founding of SofTech in 1969 because we did no further graphics work after, leaving MIT.

So the interesting thing is that I sort of span all the time of all the panelists on this session and the next, which means I have too much to cover.

It also means that another problem that I have is that this has sort of been a year of nostalgia for me, because in addition to this meeting, we had Project Mac's 25th Anniversary on time sharing this year. The Computer-Aided Design Journal asked me to do a 20th anniversary piece for them and I submitted the Statement of Objectives for the Computer-Aided Design Project that I did in 1960. All they asked for was something that the information was my words and still up to date, and they thought I would write a special thing.

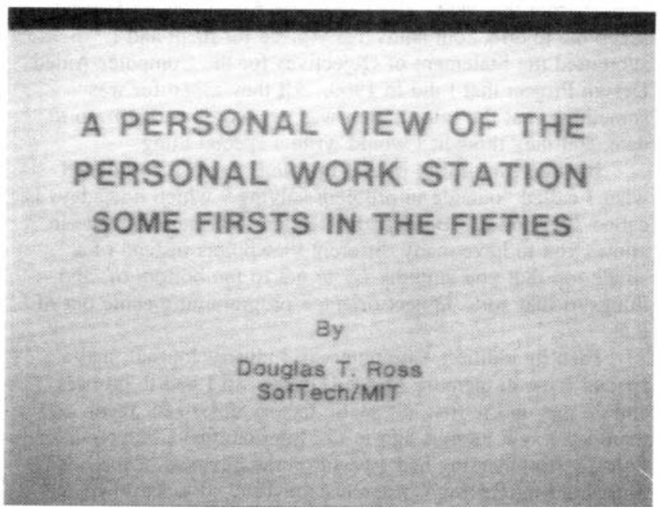
It's interesting that in that opening memo I introduced what I called "outside-in problem solving," which nowadays is called "top-down." But I prefer outside-in because outside-in allows you to have many different viewpoints instead of a single top that you stupidly try to get to the bottom of, and things of that sort. Object oriented programming came out of that.

Then in addition, Computers in Industry Journal, had a special issue in memory of Joe Hatvany, so I had to go back and do one on the first automatic factory design for them, and then just a few months ago in the International Conference on Software Engineering had a panel on the pioneers of the NATO Software Engineering Conferences in 1969. I went down and did one for that. And then just a few months ago in May, the Smithsonian Air and Space Museum just opened a brand new huge beautiful gallery on Computers in Aerospace. So I'll show you a few things that I contributed to that. And so now we have this for me to do. So I'm sort of rattling around in the old days and having a great time with it.

Mike has already mentioned the two formal history papers that I've published and I have some slides that I used in various spots there. Those are the clear ones. I tried to make more slides covering different territory, you see, for this talk, but I tried to solve a problem where sometimes my flash would make a bright spot. So this time I discovered that I had a little wire to hold the flash off to the side and I thought that would solve the problem. All it did was make them too dim. So I hope you can see what I've tried to do.

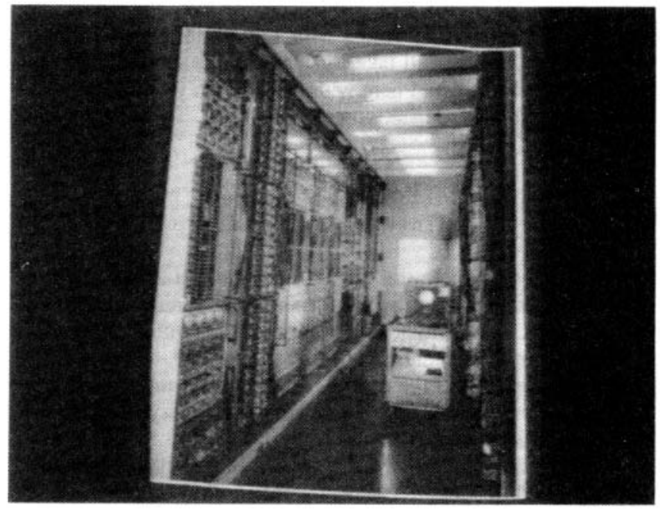
Also, when I do these historical things — hysterical things, you see — I find that my memory is very, very faulty and so I fortunately have what I call my "daily resumes," which were back in these days when I was heading up these projects. The APT and CAD Projects had one Air Force sponsorship, but it was actually a combination of about five or six projects, all going in parallel on both hardware and software matters, and I couldn't keep track of things, so I would take Polaroid pictures of the blackboard and then say what we had talked about into my dictating machine and my secretary would type it up. So I still have hundreds of pages of these, you see. In going back through that to pick out the things that had to do with graphics for this meeting, I unfortunately started where there was already a mark labeled "graphics," and it was just the first place where Ivan Sutherland turned up on the scene and there were many graphics references earlier than that that I would like to try to thread through, but I may not do too good a job of that.

So as a general theme for what I'm trying to convey and what actually drove me and my very industrious and creative project members over all these years, is that it's actually the one that Nick Negroponte was talking about in his opening remarks today — that there is much more to it than pictures. It has to be a picture language. There has to be meaning there, and the meaning is useful. You're trying to solve problems. So it really comes down to man machine problem solving. Better means of communication and expression is what always has driven our work. Even though we started in the days when the equipment itself wasn't able to do very exciting visual things, we were always concentrating on this matter of communication and meaning.



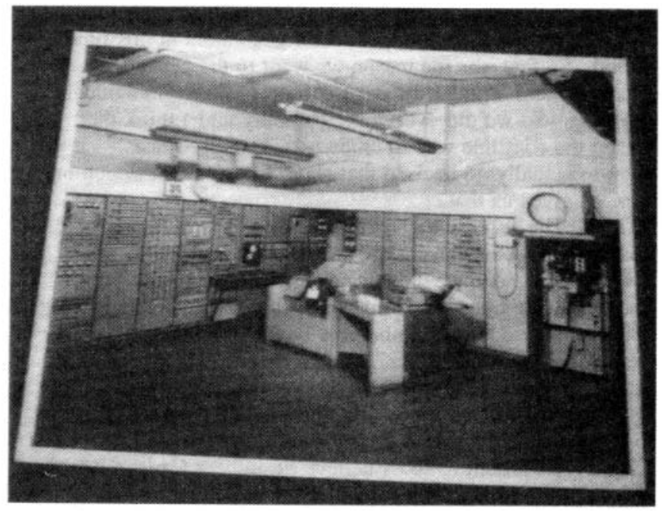
— ROSS - SLIDE 1 —

So let's get started with the slides. As I say, this was the paper that I had in the workstation conference. Now they had the word workstation as one word and that is the way it's most properly done. But I pointed out that the important thing was it's the personal work that you're trying to support — not the workstation. So that's a very purposeful blank in there.



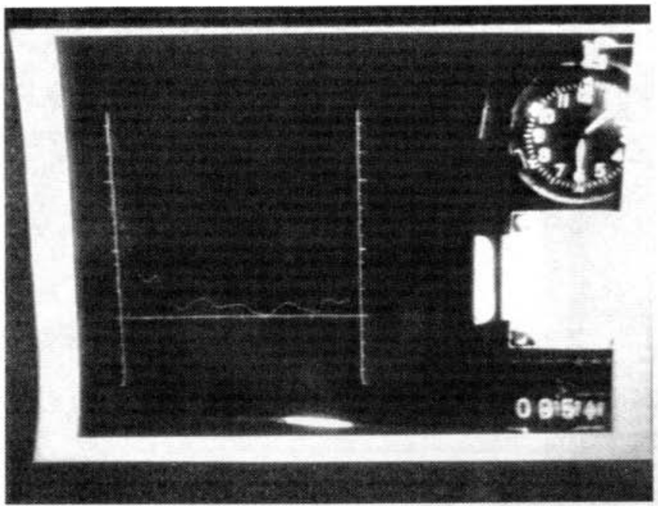
— ROSS - SLIDE 2 —

And just another shot of the arithmetic element of Whirlwind. Here is a little later shot of the control room.



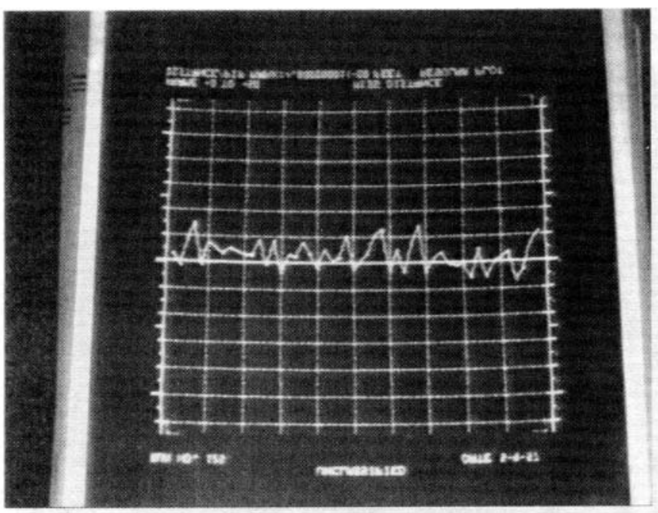
— ROSS - SLIDE 3 —

The reason I show this is that I want you to see up there on the right-hand side. There is one of the display scopes and in fact this was right after they had first shut down Whirlwind and put it and then brought it back up with (I believe it was) 24 display lines in the Barta Building, running this Cape Cod system, and when they ran the computer from the control room console here, they would slave all 24 display lines onto this scope just to be able to monitor what was going on, and they could select which one they put on it too. And below it is another scope, a duplicate, with a 35 millimeter camera mounted on it.



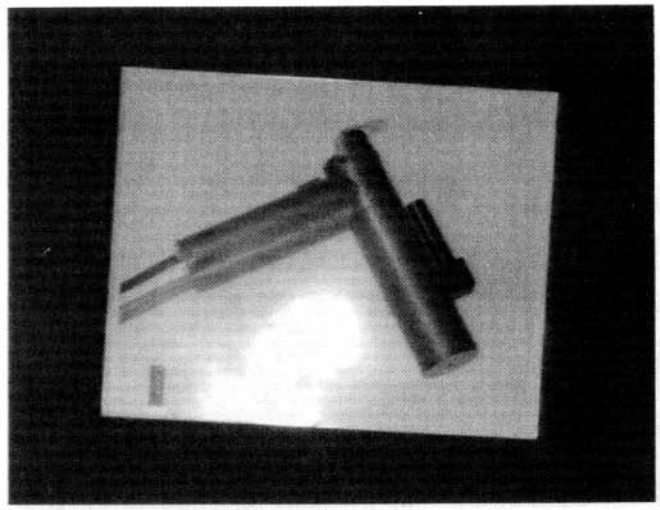
— ROSS - SLIDE 4 —

Now I'll say a little bit more about that in just a minute. But here is a picture of a graph being plotted.



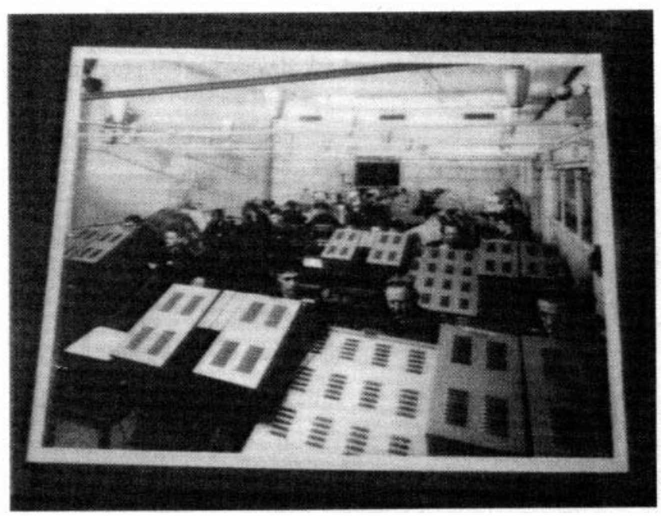
— ROSS - SLIDE 5 —

That's the scanned analog vector, sweeping the beam, that allows it to make the lines and the little tic marks on the scale, as well as the connections of the graph, and these characters are the reason they went to the Charactron. These are all made out of little sweeps, you see, and it was not very efficient.



— ROSS - SLIDE 6 —

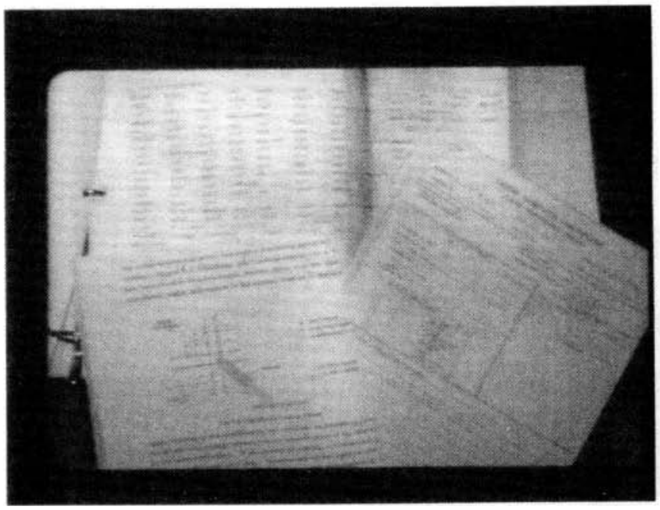
Here is the intermediate stage of that light gun. You actually hold it in your right hand with the barrel coming back up over the back of your hand, and if you look carefully at the top, you can see there's a little Plexiglas circle that's the target that you're actually going to pick up. It has a photo multiplier tube inside and a wire coming out, going to the computer.



— ROSS - SLIDE 7 —

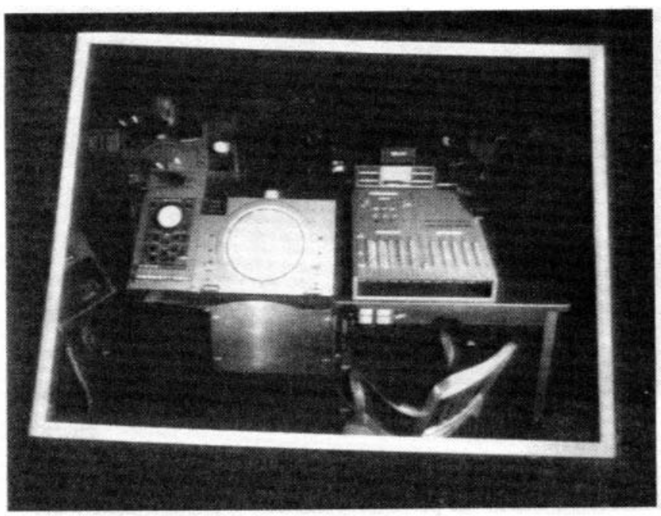
Here is another shot of that Cape Cod Room 222 — ultrasecret, and in the personal workstations paper I make a big thing out of when I first went in there, how exciting it was. But way in the back corner is not a light gun, but a light cannon, which is a photocell mounted on a tripod over a horizontal tube and they used to put a circle of yellow Plexiglas in the middle, leaving an annular region around the outside, so that when radar tracks came in, it would just see those and not see all the clutter in the already assigned tracks in the middle. And that was the light cannon that was used in the Cape Cod.

Well, I wrote the first program in the world for hand-drawn input to a computer in 1954, and using that light cannon, and here is the actual program and the way it worked.



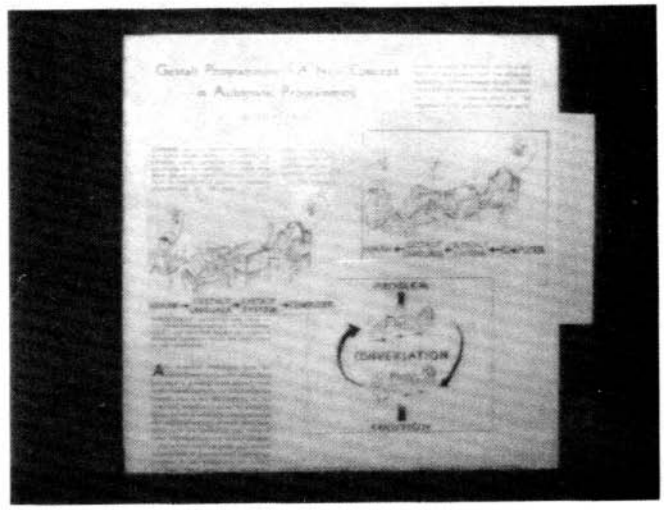
— ROSS - SLIDE 8 —

I put out a 5 by 5 array of dots as a little matrix, and whenever you see a dot, you put out the next one. If you don't see the dot, you recenter the array. So that makes it a little hill-climber from the left. So it will track any shadow that you want to use as a pointer. The program worked the first time and I wrote my name on the scope and the fellows out in the control room — see, I was doing this off in the secret room and running the computer from there. They were very upset when they saw my handwriting coming out and asked what kind of program I had there!



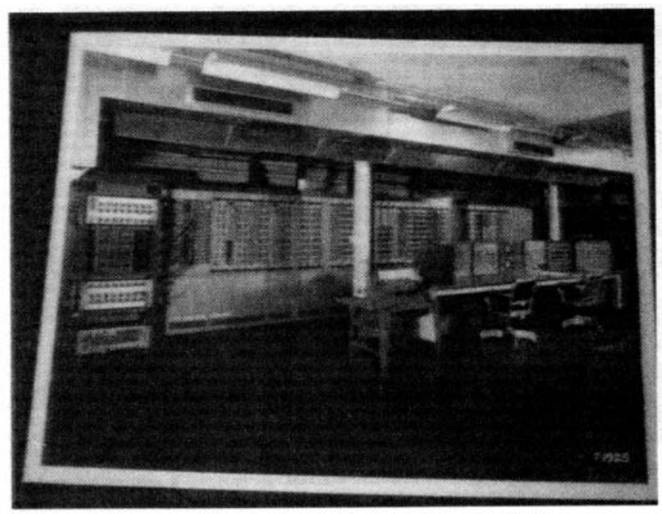
— ROSS - SLIDE 9 —

So that's what actually started off, I think, the computer graphics input for general purpose use. Here is one of the Cape Cod workstations that we used then for general purpose man machine interaction, later putting a Flexowriter in as well.



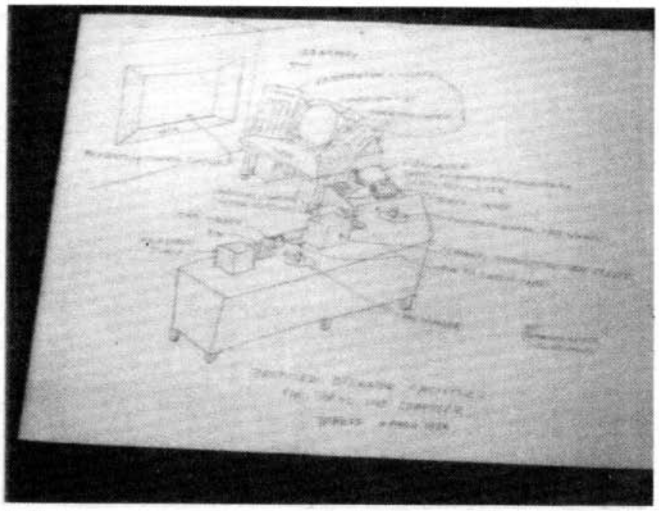
— ROSS - SLIDE 10 —

In 1955 — my first paper kicked off the Western Joint Computer Conference in California on "Gestalt Programming," and as you can see, the key thing was the man machine conversation, and that's been the theme all the way through.



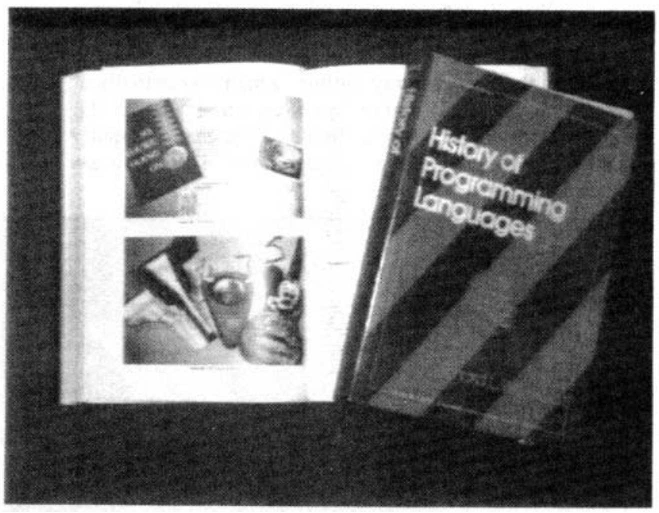
— ROSS - SLIDE 11 —

Just a quickie — here was the Memory Test Computer out at Lincoln Lab. Norm mentioned that, and that was used also for testing out the Charactron.



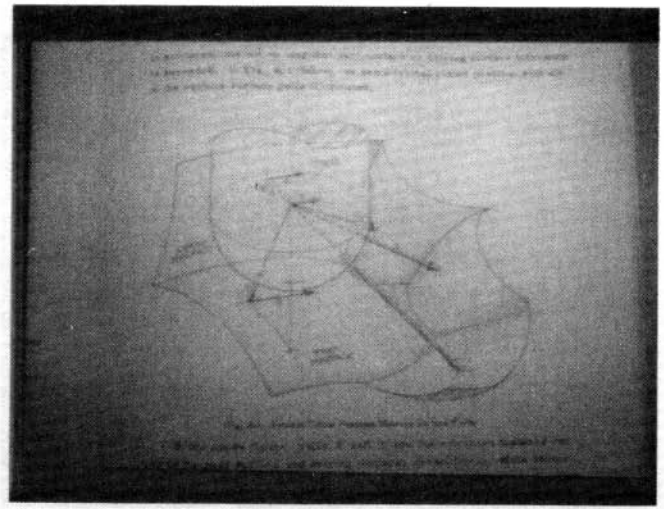
— ROSS - SLIDE 12 —

We built a man machine console and got it installed down in Eglin Air Force Base, attached to an ERA 1103 computer (serial number 3 I think), that had a Charactron tube and duplicated what we were doing at MIT.



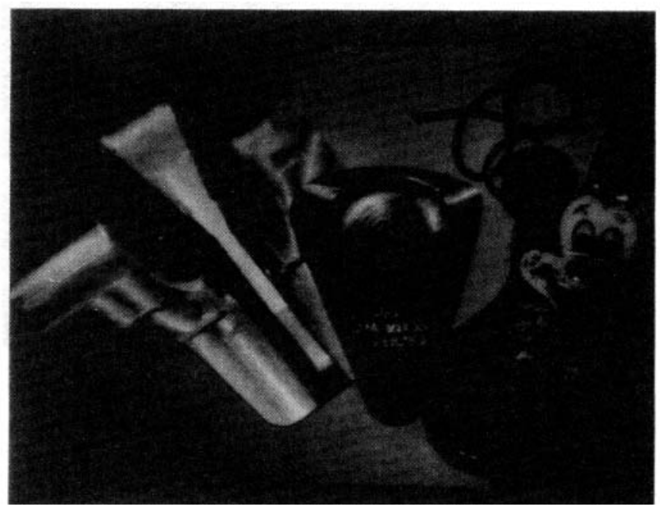
— ROSS - SLIDE 13 —

With respect to APT, this was the History of Programming Languages book that came out from that, and I guess I don't really need that.



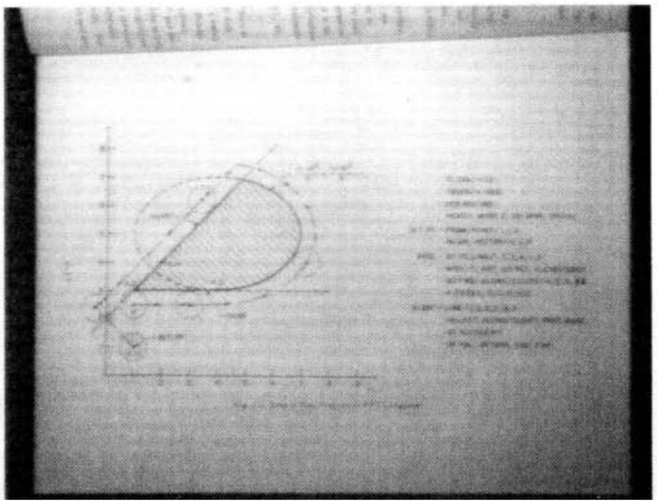
— ROSS - SLIDE 15 —

Here is the problem though. You have an arbitrary-shaped spinning cutter moving around in space and it has to keep track of a minimum distance to a whole bunch of different surfaces, one of which is the part you want to cut, another is the driving surface that determines by its intersection, the space curve which will be the three dimensional motion of the tool until it bumps into one of the other surfaces as the check surface, at which time you have to start another curve, you see.



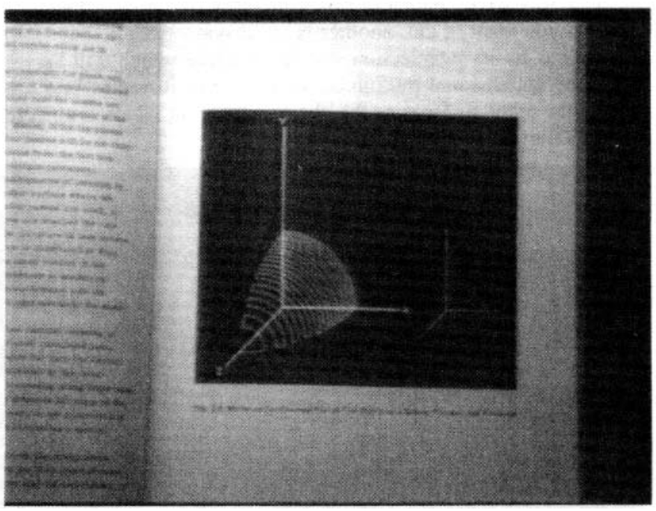
— ROSS - SLIDE 16 —

Very, very complicated calculations and here are the artifacts that actually appear down in the Smithsonian Museum — including the Mickey Mouse guitar. The ashtray was a souvenir we gave out at a press conference when we turned this over to industry. The system was developed jointly by — it ended up with 19 different aircraft companies providing programming staff to join with my staff and we did it all over the country and put it all together and it became an international standard. The other part is an engine mount for the F-100 aircraft, done on the MIT milling machine.



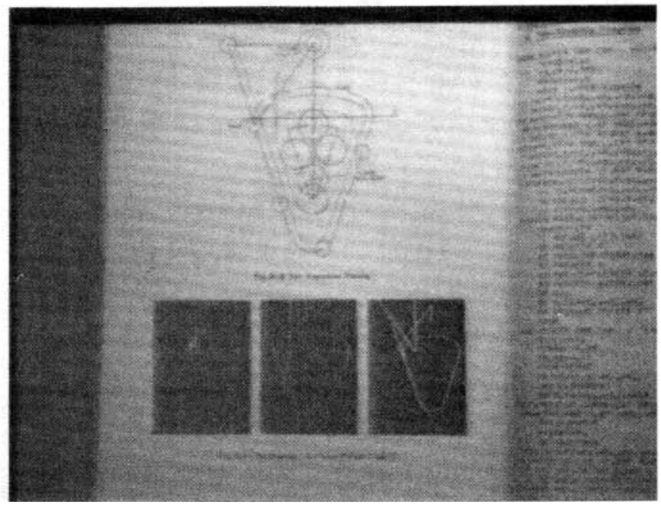
— ROSS - SLIDE 17 —

Just a little picture of the kind of language it was. It's a very simple English-like language with a few commands and that's one of the test parts.



— ROSS - SLIDE 18 —

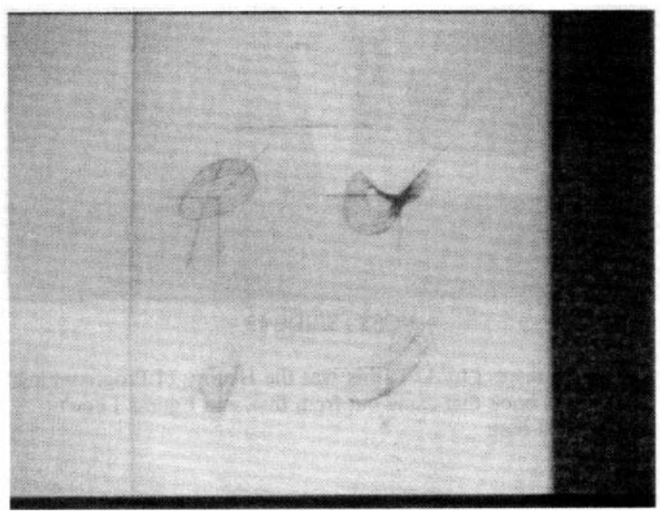
These are the Whirlwind axonometric displays of the cutter center path when you index a driving surface across a spherical part and a pyramid part made out of planes.



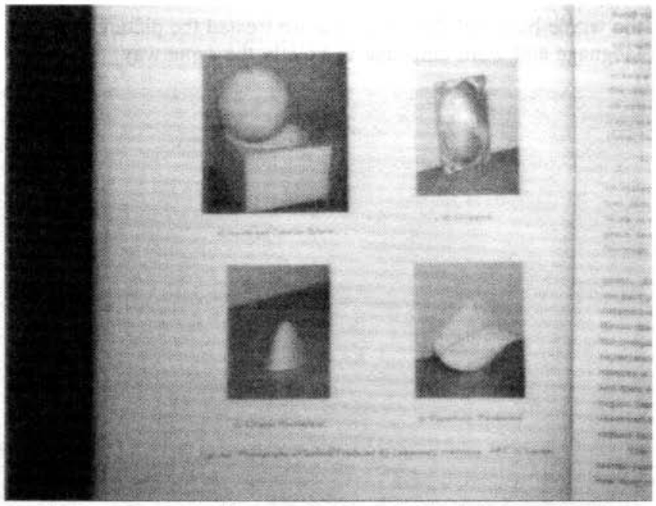
— ROSS - SLIDE 19 —

This gets pretty faint, but this is from the test programs and again used in the press conference. You see, the language allowed you to pass circles through intersections and tangent and so forth so that you could parameterize a whole family of parts and then by just changing the the values of the parameters, distort the shape. So at the lower portion I hope you can see it well enough.

Here's the same ashtray outline, which was actually a "rocker arm cam part," and as you change the values of the parameters the shape distorts. Some of the early computer graphics-like things that are so much more elaborate today and very crucial.

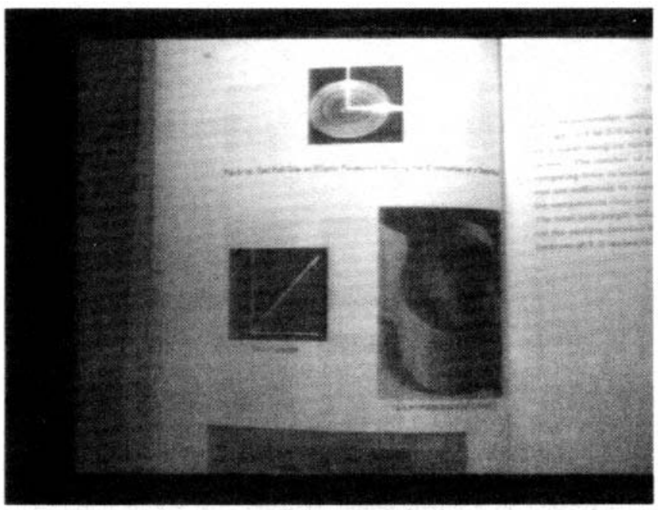


— ROSS - SLIDE 20 —



— ROSS - SLIDE 21 —

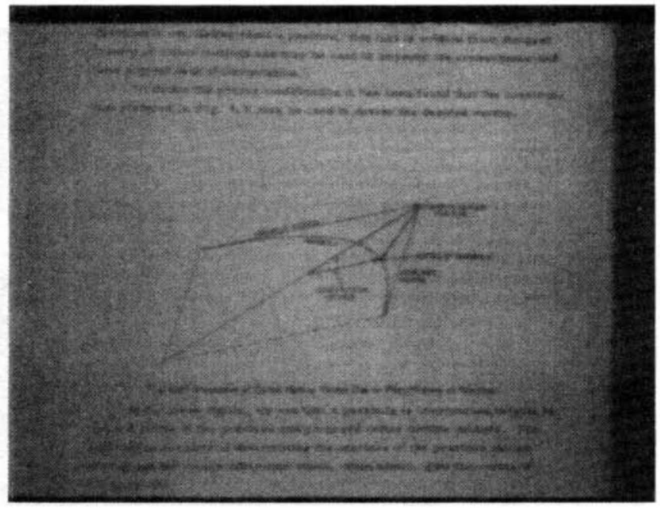
What I've been talking about is a system called APT-II which was a two dimensional — it was three dimensional space curves. But we also had an APT-III which was three dimensional surfaces and so with automatic cutter-path determination. So here's a bunch of APT-III cutter paths and here are the actual parts cut out of either aluminum or styrofoam — ellipsoids and paraboloids and elliptic paraboloids and things like that.



— ROSS - SLIDE 22 —

I threw in this one. I don't know whether it comes out well. This is one of the new shots where the flash wasn't quite enough. But the vertical one there was cut out of a piece of cherrywood, an elliptical paraboloid shape, and it was so intriguing to people that they ended up actually putting a pair of these together and for a while claimed that we were sculpting a model of Marilyn Monroe. What has happened to my slides here? Is there something there? No? Can you find out why I have no more slides? Is that the hook?

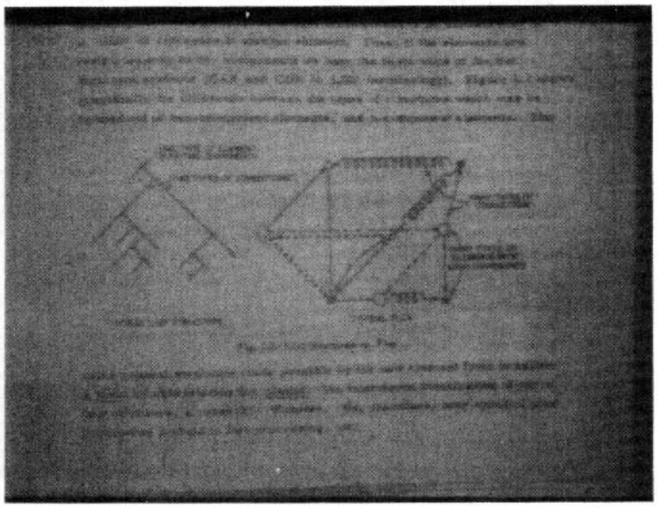
We have lots of them.



— ROSS - SLIDE 23 —

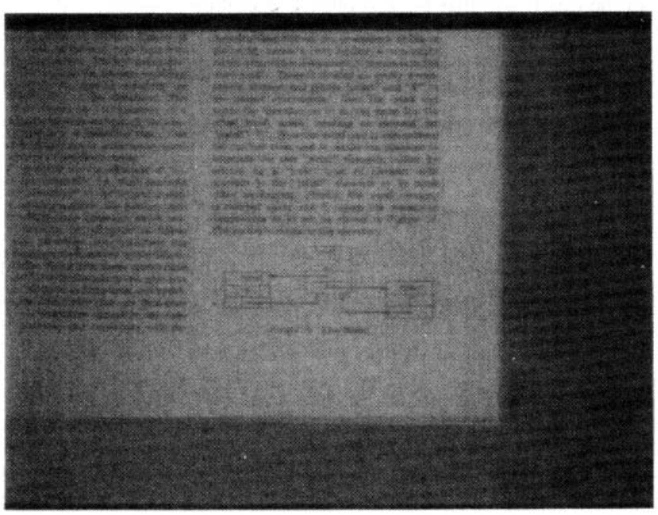
I can't see them on this — tell me what's there. Let's see; maybe I've got a list here. This is a very important point because it's at this point that I wanted to mention the precursors to the very famous and very influential Steve Coons' "Coons Patch," because this is — I hope you can see it; I can't see it against the glare. But it's a system that I put in to greatly improve the speed of convergence of their very complicated calculations that went into producing successive cut vectors in the APT system. It's a technique I call "parabolization," and if you have an iterative computation going and these successive vectors go from one state to the next state to the next state and you're trying to converge in on a finish, a final state, well, because it always has errors of round off and so forth in the computation, no one is going to be correct. So what I did was I put a linear blending of successive iteration vectors from one to the other, and that sweeps out the envelope of tangents of a parabola, if you think about it. So instead of going from the first vector to the second vector's end point, you go from the first vector to the apex of the parabola, and it has the neat effect of giving very high order convergence to any iterative computation.

And I remember talking about this when I first got together with Steve Coons, who along with Bob Mann, started off the mechanical engineering portion of the computer-aided design project. We subcontracted some of our contract money to them for a matter of five years at the beginning, with very fine results. So I was talking about this and it reminded Steve of his work that he had done when he was, I believe, at Chance Vought, doing airfoil shapes, and sure enough, it wasn't right away, but a few years later out came this very marvelous series of developments of Steve and his Patch.



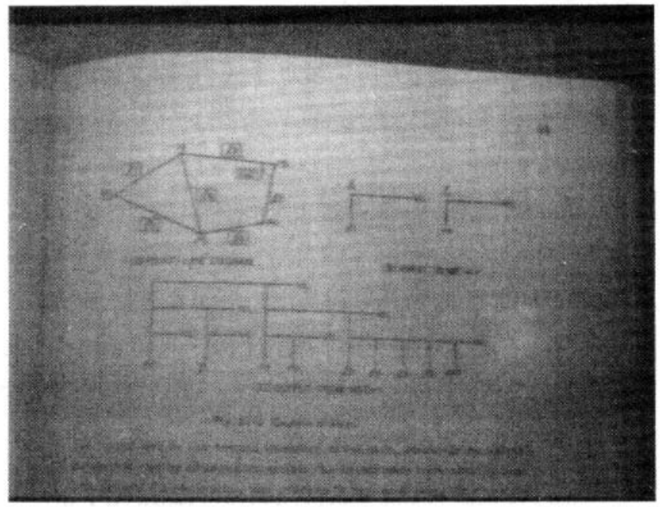
— ROSS - SLIDE 24 —

Now the next one here — again, in order to do this thing of having meaning behind pictures and having them be language elements, it's extremely important that you have an actual modeling of the reality that's being expressed or represented. One of the principal offshoots of the development of the APT System was we had so many different surfaces to keep track of and so many different segments that made up the boundary of a tool, that I invented what I called the principal of "reverse index registers," which turned out to be an efficient way of implementing pointers. So we had what we called "beads and pointers." Nowadays they're called "records," and are of course, the sine qua non all software engineering languages. But this was in 1959 and we had no high level language for it. APT itself was a high level language in those days.

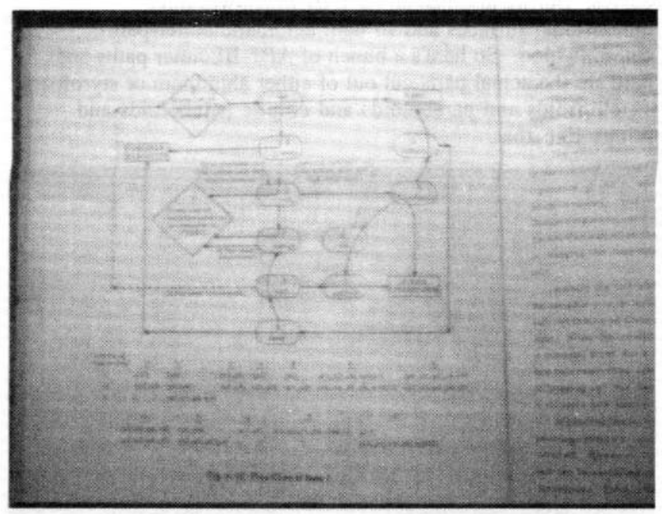


— ROSS - SLIDE 25 —

So this was where the storage of information fit in. Now we used it for both geometry lines. Here is an actual model of a line, as well as entire problem modeling, including language processing. I have a bunch of slides here which I will just rip through very quickly now. But the point I wanted to make with the whole bunch of them was that we treated the picture language and word language in exactly the same way.



— ROSS - SLIDE 26 —



— ROSS - SLIDE 27 —

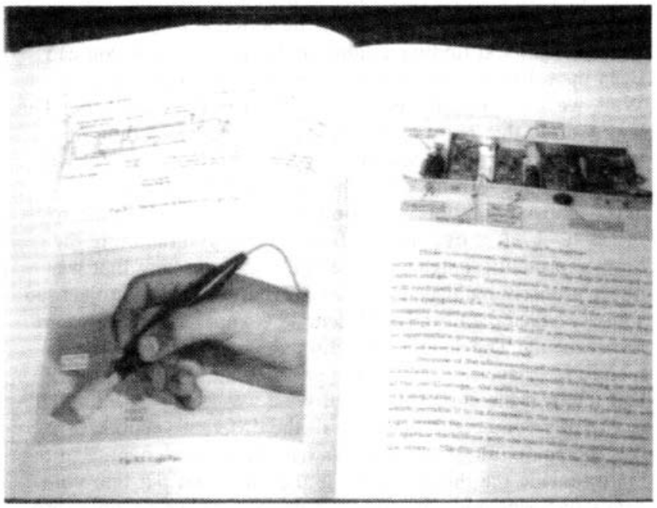
In fact, we called it the Bootstrap Picture Language, and from this earliest Point Line Diagram Study in the 1959-1960 period (when we actually used the Lisp programming system, first version, as a programming vehicle), that got us into this fancy data structuring and we were carrying along the problem of constraint satisfaction and so forth in parallel. We started earlier and had it underway, but when Ivan got in and made his second version of his Sketchpad program, (the first one essentially duplicated what a draftsman could do and was sort of like our APT definition of preprocessing with lines and circles), and then later when the constraint satisfaction idea came into the picture for him, is when he really took off.

So we ran in parallel, doing similar developments with this making of language — picture and word language being processed the same way. And in fact, now I've gotten up to the Spring Joint Computer Conference.

This slide is just a list of the papers we gave in the 1963 Spring Joint Computer Conference Special Session. You see Sketchpad is listed there and the first paper on Theoretical

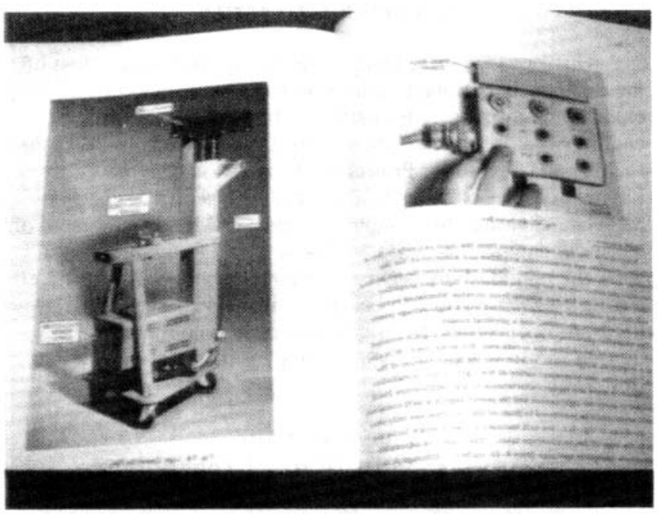
Foundations for the Computer Aided Design System is this business about how to make it all meaningful and work together.

Let me see if I can get to a TX-0 picture. This is one shot that I have from TX0 which moved down from MIT to Lincoln Lab. As you can see, the writing of words there, but the upper left corner of it is the mouse in a maze solving program that every once in a while still gets run down at the Computer Museum here. I did that with John Ward.

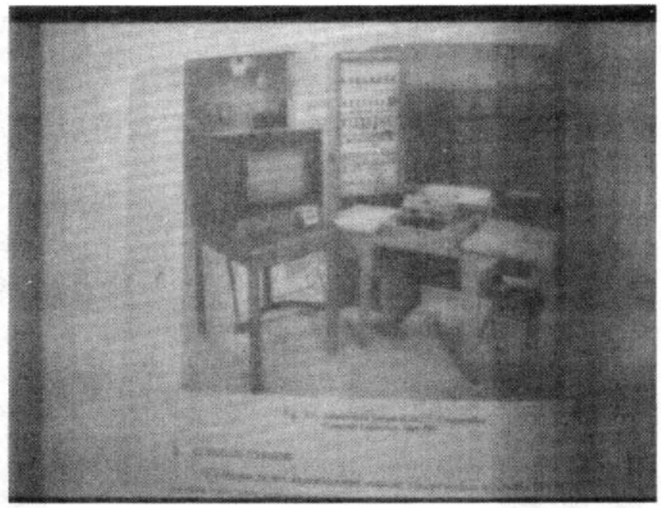


— ROSS - SLIDE 33 —

Then these are efforts to refine the light pen technology. This was the light cannon and push buttons that we were putting on the IBM 704.



— ROSS - SLIDE 34 —



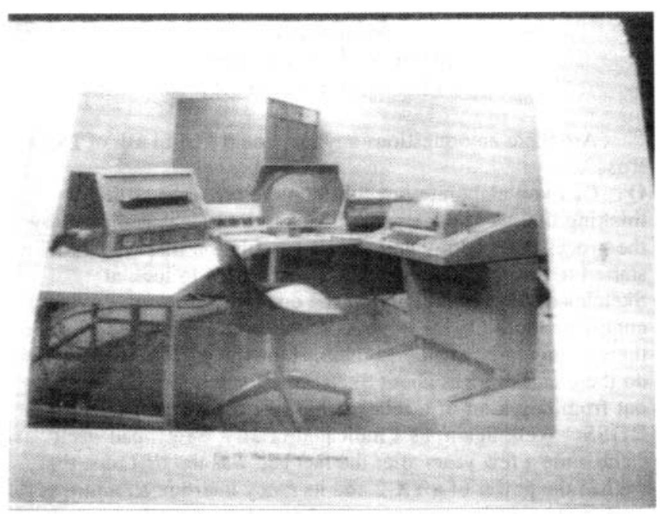
— ROSS - SLIDE 35 —

This, interestingly enough, is where the name "Kludge" first came into use, it was the way we were "kludging" things up to hook them on to the IBM computers, and it was only later that the real Kludge — which I will show you in just a jiffy here was built. This is more 709 Kludge.

This is the Coons Patch as applied for ship design, and I actually was the one who came up with the idea of using binary rate multipliers, which were the method by which the machine tool was controlled, building them into get real time rotation and scaling in the display console itself.

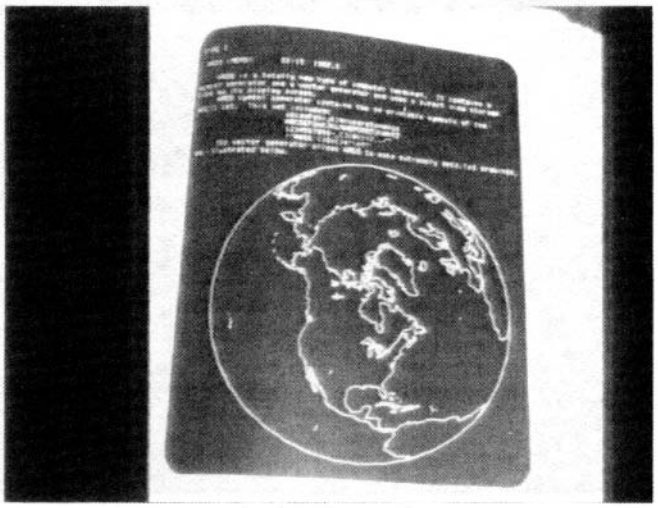
This slide is supposed to be the simulation of clipping that Rob Stotz did as part of his thesis before, building the Kludge later. The simulation was done on the PDP-1 computer.

Here is a wire frame drawing from Tim Johnson's Sketchpad 3 I think. And finally, oh, here's the Kludge.



— ROSS - SLIDE 40 —

That's the ESL Display Console, complete with teletype. That's one of its later pictures.



— ROSS - SLIDE 41 —

Yes, this is the final — is this the ARDS display? There is the ARDS display — fine. And with the World View, the ARDS display was the application of the same display technology but using again the storage tube which by then had grown up to a sizable size and was very good for use over phone lines in a time sharing environment.

So as I say, it's a little hard to cover — even in 20 minutes — 17 years, and I've left out lots and lots of things that would be important to say. But I guess the key thing is that using the the resumes,I will continue to write more formal things and if you become an aficionado on these things, there will at least be these original sources of what we actually were thinking about, as we did these things in those days. Sorry to be so rushed. I hope it's been interesting. Thanks.

Moderator
Michael S. Mahoney
Princeton University

Are there any questions anyone would like to ask of Doug Ross?

Q. Can you elaborate just a little more on some of the thinking that went underneath. Many good ideas come out by the process of analogy and there's speculation that when APT started to develop and then when you started to look at Sketchpad ideas essentially on the one side you were driving a controller tool. On the other side you were driving a beam. So there seems to be analogies there. Can you just elaborate? How do these earlier ideas about Sketchpad come out? Do they come out from any kind of a relationship like that?

ROSS: Well, again, as I indicated briefly, Sketchpad itself came quite a few years after the fact and was the first time that we had the power of a TX-2 and its fancy interrupt structure as well as its very large memory and speed, to actually bring these things to life.

I had tried to get probably three or four or five students along the way to do Master's theses on using the light pen techniques and so forth, but the equipment just wasn't there well enough. We had these tests going, you see, but to bring it all together was not there. But I think the key thing was that all of our work in my projects — both APT and Computer-Aided Design and there were other things that had to do with the doing experimental programming environments and operating systems, that sort of thing. The key thing was we always wanted to have real problems with real users that needed us to give them a tool they could use and perform. So fortunately, all these things fitted together that way, and so if we — I've had people accuse me still that they ask me a simple question and I abstract it up two levels, answer it at that level and bring it back down and they can see that yes, it has something to do with they asked, but it's really more, you see. But the key thing that we learned very early in the game was that you don't solve very difficult real problems by looking around for the special cases and finding a chink in the armor where you can open them like a clam, which is the way you solve puzzles. Well, we're not puzzle builders; we're technology builders. The way you do that is by getting this global understanding of which the particular problem is a special case. So I think all the way through, from APT, where we used to call it a "systematized solution," right on through — I was going to mention. I left it out entirely here — that programming the bootstrap picture language developed into a system that was never properly published because we shut down the project too soon — called the CADET System — Computer-Aided Design Experimental Translator. That did indeed process word and picture language in exactly the same way and built up models that retained the entire history of what was being built and all of the ways you could get it — all the different versions of it.

So again, all of these things have to do with capturing the reality inside the thought process that then you put into your data structuring and your program design. That's a long answer again — probably more than — I've done the same thing again you see.

Moderator
Michael S. Mahoney
Princeton University

Thanks very much, Doug. The last speaker in this first of the two panels is Robert Fano, who has a short but very eloquent description of his career — trained as an electrical engineer first in Italy and then in the United States. He was the founder and director of Project MAC, which is now the Laboratory for Computer Science, and also served as the first associate chairman for computer science in MIT's department of electrical engineering when MIT was trying to figure out just what the relationships between computer science and electrical engineering were going to be. He retired several years ago as Ford professor of electrical engineering and now occupies himself, he tells us, with some complex financial computing. Bob Fano.

Robert M. Fano
Massachusetts Institute of Technology

Project MAC. Oh boy. The decision to start Project MAC was made shortly after Thanksgiving 1962 and the actual effort started — or at least we were allowed to spend money in April '63. Now I would like to give you some idea of what the state of computing was at that time and particularly the man machine interface. The goal of Project MAC was to foster machine-aided cognition through the use of multiple access computer — namely time sharing computer.

So the state of the technology at that time was very important. The queen of scientific computing at that time was still the IBM 7094. Every time that I sit down at the MAC II next to my desk, I can't help thinking that that small piece of hardware sitting next to me is significantly more powerful than the 7094 in all dimensions, except, thank God, it doesn't have tapes. Now the state of the art at that time was rather primitive, as you gathered from the previous speakers. The new IBM machine, the System 360, came in the next year in '64.

At the time that Project MAC got started, essentially lower case characters were not used — at least commercially. The IBM 1050, I believe was the name, which was a computer adaptation of the golf ball office typewriter came in shortly thereafter. I was with a number of other people to the Teletype Company in Texas in late '62 or early '63, trying to persuade the company to put out teletypes with lower case. Well, the chief engineer was rather opposed to it. He said we made a market survey and the market survey came in saying that nobody knew what to do with lower case characters, so we're inclined not to do that. Well, we persuaded him to do it and Teletype came out with the — I think it was Model 37 Teletype and then the model 33 which was a cheap version of it.

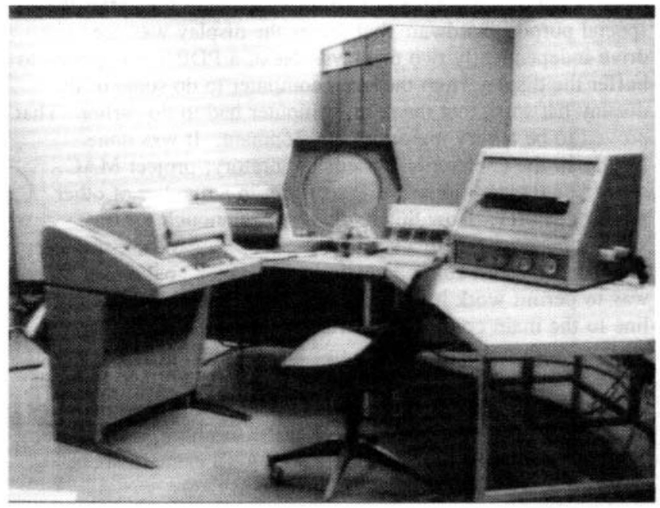
You heard about the early days of graphics. What was not said is that the work that was done — and you will hear more about it by Ivan Sutherland in Sketchpad — kept the whole TX-2 computer busy and Ivan Sutherland and other people doing that sort of work signed up for the TX-2 for hours. In 1963 hackers were beginning to cut their computer teeth on the PDP-1. It was the only commercial machine built for graphical display. Now for us, who were concentrating on man machine interaction in the context of time sharing of a large computer, graphics presented a serious problem. Not only graphical work required a lot of computer horsepower continuously, but also was very memory limited, which was a serious problem for us.

For you that don't know, the IBM 7094 had normally 32,000 words of memory or 36-bit words. TX-2 was very rich. It had 64,000 36-bit words. Our 7094 for time sharing had two cores memories of 32 kilowords each. One of them was dedicated to the operating system.

Well, in that situation Project MAC got going, the time sharing system got into operation in October '63 and it was an exact copy of the time sharing system developed by Professor Corbato at the MIT Computation Center earlier. And it was in operation in October as I said.

With respect to graphics, it was just sheer luck that at about that time a very sophisticated graphical display to which Doug Ross referred, was being completed by the Electronic System Laboratory, also at MIT, which had been engaged in the computer-aided design project for quite a while. And that display — which I will show you in a minute — was first tested on the Project MAC machine and without time sharing and then the addition to the supervisory core program was written and in early January it was operating under time sharing.

Now let me show you what it looks like. Incidentally, the group, the hardware group that developed this machine, this display that you see now, and the one that I will mention later was headed by John Ward who unfortunately passed away about a year ago. He really should have been a member of the panel instead of me. John was the man in the Computer Aided Design Project. He handled the hardware and Doug Ross handled the software.



— FANO - SLIDE 1 —

The idea here was that since machine power was really scarce, one could develop a display system with specialized hardware to do the display work — to minimize the work of the computer that was in the background, and that's what was done. I won't go into detail on how it was done, basically, it was an incremental vector generator.

Now the most outstanding thing of the display, you will see that globe in front. That globe is a joystick, in effect, that was used to rotate the image on the screen. That is, a three-dimensional object was defined in memory and with the globe you could control the speed at which the object appeared to rotate. The goal was to provide a three-dimensional feeling for the object. And that worked very well. Now, that display was immediately very popular and a lot of very interesting work was done with it. You will hear about Steve Coons' work later on. I will mention one application in biology that was very, very interesting and was the original piece of work that led to a lot of other things later.

What happened is that Professor Cy Levinthal, a biologist who was at MIT at that time, came to see me — I think it was just before Christmas 1964 — to find out something about some computation that he wanted to perform. He was interested really in the three dimensional structure of proteins and other large molecules and I thought — ah ha, this is a wonderful opportunity for man machine interaction.

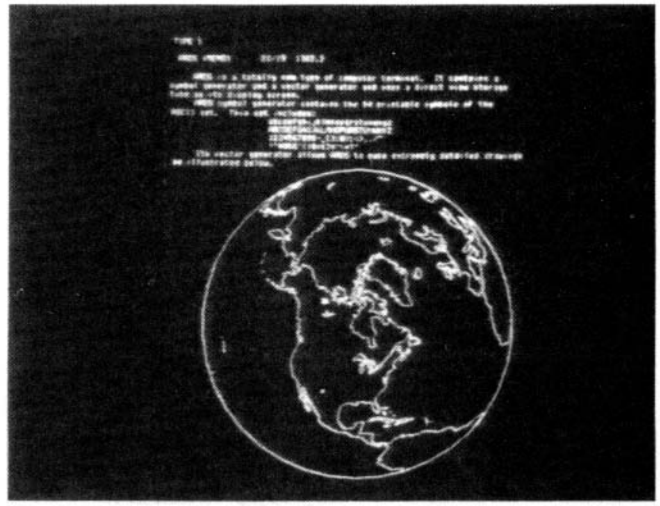
So I said I'll talk with you about that a little later. Let me show you around here. So I took him to the display — affectionately known as the Kludge — to demonstrate it to him. He sure caught fire. I was going skiing right after that but the next morning he called me and said how can I get to work? So I got my assistant director — someone you may know, Dick Mills — to get him going. And by the time I came back from skiing after New Years', he was programming, and he really caught fire.

The ability to view a molecule in three dimension was really the thing that appealed to him. Well, his work went so well that it was not long after that that he got his own computer and shortly thereafter he left MIT and went to Columbia, and the last I heard he was building a super computer specialized for molecular work.

There were quite a few applications similar to that that opened up with the availability of the Kludge. What you see there is the original Kludge I believe. Quickly thereafter the special purpose hardware that drives the display was used to drive independently two displays. Next, a PDP-7 was gotten to buffer the display from the main computer to do some of the display list work that the main computer had to do earlier. That proved to be a very important development. It was done mostly in the Electronic System Laboratory; project MAC supported the additional work, as it did in a number of other projects at MIT. Now this really was not enough for time sharing because this display had to be very near the computer. You needed a fast communication line and the goal of our effort was to permit work by any person connected by a telephone line to the main computer. Certainly you could not have a display of this type. Well, in my thinking I kind of set a goal. I said what people would be willing to pay for a capable display that you could use through a telephone line was something in the order of magnitude of the cost of the furniture in an office, and I kind of made a quick computation — it came to about $5,000.

— FANO - SLIDE 2 —

So the Electronic System Laboratory, and particularly Rob Stotz, got busy trying to solve this problem and a masters thesis by Tom Cheek was done on a display that would meet the purpose. Now the big problem was how do you store the image so that you can display it flicker-free. Obviously you could not reproduce it flicker-free through a telephone line. So the only choice was to use a storage display tube where you could generate the image and let it sit there. Then you could erase it, but you could not have fast interaction. But this was better than nothing. So let me show you.



— FANO - SLIDE 3 —

This is the ARDS display and what you have there is a storage display tube. As you see, you also have a mouse sitting there, and a keyboard. And the sort of picture that you could get — this is the same picture that Doug Ross showed you before — you could get a picture with a lot more detail than you could get on a Kludge. But if you wanted to change it, you had to kill the picture and restart over again. And to kill the picture — if I remember correctly — took 30 seconds.

So it was not very convenient for man machine interaction but that's the best that you could do. This used a Tektronix storage display tube and Rob Stotz tried very hard to persuade Tektronix to manufacture this display, but he didn't succeed. So he went in business for himself to produce this display. It was not too successful. Perhaps Tektronix was right in its marketing. I don't remember now the name of the company. Doug, do you remember the name?

ROSS: I don't recall (INAUDIBLE)

FANO: Well, anyway, it was bought by Adage later on and I don't know what Rob Stotz is doing these days.

This is most of the work that we did in Project MAC on display. It wasn't that much and it was all done in the Electronic Systems Laboratory. Basically the technological basis for significant display work under a time sharing operation just plain was not available until late in the '60s and the early '70s. That's when raster scan display became available and one could begin to think about on-line editing and all that sort of thing — on a display. We were of course doing on-line editing on a typewriter, but that was a different sort of a business.

Thank you very much.

Moderator
Michael S. Mahoney
Princeton University

Any questions specifically for Bob Fano? In the back there.

Q. (INAUDIBLE)

FANO: I didn't quite hear — did Tektronix —

MAHONEY: Use the microphone behind you or this one.

Q. Didn't Tektronix dominate the computer graphics market for a while with their storage tubes?

FANO: I really don't know the details. It's just — he tried to get them to produce the display but they were not interested. They produced the tube.
Q. They did produce tubes quite successfully; they made a lot of business. That's why they're in graphics today.
MAHONEY: Can we turn on the mikes on the panel?
ROSS: Anyhow, as I recall, one of the problems that Rob Stotz had with his company was that Tektronix essentially went into competition with him, and yes indeed, they did for some time make and sell quite a number of systems based on the same tube. I think also that the Computek that was started by Don Haring and Mike Dertouzous also used storage tubes in their early displays — didn't they? Remember that company?
FANO: Yes.
ROSS: It's a little hazy in my mind, but yes, Tektronix was a leader for quite a while.
FANO: That's right; I had forgotten that. Tektronix, after saying no, we are not interested, went in business doing that same thing. That's right. You are correct.
Q. Where did the name Project MAC come from? I've heard there's some stories.
FANO: The name Project MAC. MAC stands for Machine Aided Cognition and Multiple Access Computers — one being the goal and the other being the means to the goal. Lots of names were coined afterwards. The one that I remember very well obviously coined on the West Coast was More Assets to Cambridge. I also have to tell you that by April, 1963 the name of the project had not been coined yet and I was forced to coin it in a hurry because the first employee, my assistant director, arrived on the scene at MIT and needed a parking sticker. And the office that distributed parking stickers asked him where do you work at MIT. And there was no name. So they couldn't, they wouldn't, give him the parking sticker. So he came back to me and that evening I coined Project MAC. So the next day he could get his parking sticker.
MAHONEY: It all really did begin with Henry Ford. We have some time left if anyone has a general question or observation, comment. Let me throw the floor open and we'll keep our panelists' microphones open.
ROSS: As a matter of fact, I have — I was just thumbing through these resumes, extracts, and with respect to what Bob found, I was just checking your memory. It was excellent, Bob. It's on March 13, 1963. In those resumes, I referred to the project as just the "ARPA Project" at that point: "At the ARPA meeting, the 7094 is due to be shipped September 1. Hopefully we will be on the air October 1. (They made it.) They also decided to have some combination of letters to make a name for the project. The current ones being IPS for Information Processing System. But hopefully we can get some better letters, and Bob Fano will decide what it's going to be." There you are!
FANO: There was only one name that had been used up to that time. When I decided to start Project MAC for a variety of reasons that I won't go into, I wrote a little memorandum that was sent to every significant member of the MIT administration, including the dean of engineering, who was Gordon Brown at that time. Later on he told me how he instructed his secretary to file the memo. It was under FF — meaning Fano's Folly. So that was the only name. So I had to hurry to concoct some names so that it wouldn't be known forever as Fano's Folly.
ROSS: Let me read just the next sentence or two here. It's really fascinating to go over this stuff that was actually written at the time. Right after saying this about how Fano was going to decide what it's going to be, then I described our console, using the memo that Rob Stotz put together listing the specifications. "Then Fano introduced my suggestion — I had talked to him just a couple days before — that if ARPA could buy some of Corby's computer time, then we could put our console on the 7090 and make it generally available. Everybody was very enthusiastic about the console — especially since it would be in effect — a $100,000 gift. And so it looks like we will put it on the 7090." So it actually had a short time when it was hooked up to the 7090 down at Project MAC, which is why it was able to be brought up — at the computation center — which was why it was able to be brought up so fast when Project MAC got on the air. It's very interesting. Good times.
Q. I was just interested to know in that picture of the Kludge — because I was involved — got into graphics way about the early '60s myself. Was that in fact a Dec 338 display?
ROSS: Yes, it was a Dec something 30, right.
FANO: Yes, I can tell you what that something was. I think I've got it here. I've got copies — type 30. Digital Equipment Corporation — 530. Then the second one was a better model. Incidentally, one of the significant developments that took place shortly thereafter was the design and implementation of a different kind of light pen. You see, the original light pen was sensitive to light. And that through the phosphors you had a time delay involved. So the hardware developers — I don't remember whose idea it was — developed a light pen that was sensitive to the electron beam striking the screen. Therefore you avoided a delay introduced by the phosphor. That was a significant development.
ROSS: Yes, I found a reference to that in my resume someplace too. That was very interesting.
HURST: If we have some time, I would like the panel to discuss some of the feelings and the atmosphere of the areas at the time that you were working. That's been one of the most interesting and fascinating things for me to hear about. What the atmosphere was, what the students were like. Did you understand how important all of these milestones were?
FANO: I can talk about the feeling in Project MAC and in the MIT community, but that's mostly about time sharing, rather than graphics. It was a period of really great excitement. Time sharing started out as a way of sharing a computer. What it turned out was something much, much more important than that. It became the repository of the knowledge of a community. What made the system real effective was the fact that it had the first disk file — the IBM — what is it — 1301 — came in in 1963. And that made a tremendous difference. Everybody's files were stored on the disk file.

So it really was the storage of the knowledge of the community. As a matter of fact, it was not long after the beginning that we had to institute an editorial board. There were three ways in which information, programs and data, could be stored in the system. One was in the personal file of the individual. Another one was in a public file accessible to everybody. The third one was a system commands. Now somebody had to decide where to put it, and this is very analogous to an editorial board of a journal, when you're trying to publish. And it was very interesting, the effect on programmers of that. Before then programmers never thought of their work being used by anybody else really. So they didn't document it, they didn't — it just was a personal thing. Suddenly a program became in effect a publication, a computer publication, and people poured a lot more energy in writing good program and documenting, and of course they wanted it really published. Now there was also a subtle thing there. Everybody was allocated a certain amount of storage space on the disk — disk file. But the public files were not allocated. So people had an interest in being able to put their work in the public file and even more in the command file. There was a real prestige attached to become a system man.

That was a very interesting thing. The other very interesting thing is the community effect. People knew each other through the computer system, and that happened on many occasions of people meeting physically at some time and saying, "Oh, that's you!" They had communicated through the computer before and they knew about each other.

The one thing that I regret very much is that I didn't get a very good sociologist, social psychologist, to look at what was happening early enough, because of course, it is a phenomenon that you cannot recreate. If you miss that chance, you'll miss it forever. But from a sociological standpoint it was an extremely interesting phenomenon.

TAYLOR: I might speak to that idea of what was the atmosphere. Whirlwind was probably the first computer that ran well, that was fast enough to be a real time computer, and therefore when it started to run in about 1951, it was a feeling of excitement — probably more than I've ever had since.

In the year following that, when we get the core memory on the thing — as I pointed out, the storage tube was kind of cantankerous, but after we got it to the — the core memory on it, which is about that time, the first year after that we were able to do much more interesting things. We still had some displays on it, some of which Doug used. We had 25,000 visitors. We had to set up a department to handle the visitors because they were there all the time. And of course, it was tough for us to continue development on the SAGE system, which this was the prototype, with all these people walking through there all the time. We had people from Europe, Switzerland, Germany, Japan, China — all of that. And it went on and on for such a — so that all the students that came out of MIT wanted to do their thesis on this thing. So we had more students trying to do theses than we had people working on the system. So as far as — everybody seemed to sense they were on the threshold of something very important. And it was largely I think — this was a display being able to communicate with what was going on that we were able to get that excitement satisfied. It was a little dull when you just put in numbers and got out numbers. But when we got the display started, it changed the whole thing, and I think that meant not only the bandwidth you can get out of a display system, but the man machine interaction of the light pen just seemed to excite people beyond comprehension.

Moderator
Michael S. Mahoney
Princeton University

Thanks very much. We've reached the end of the first half — the first panel — taking us from the early days of Whirlwind up to the early '60s when, as Doug and Bob suggested, some of the hardware became available to realize in actuality the ideas that people had about putting pictures on the screen. We'll pick that up in the second session which will begin around 3:30. I do want to thank our first set of panelists, Norm Taylor, Doug Ross, and Bob Fano, for very interesting insights — and for some sense of what they were thinking when they did what they did.