

INTERVIEWEE: JOE WEIZENBAUM
INTERVIEWER: Robina Mapstone
DATE: 17 April 1973

RM: This is Bobbi Mapstone and I am talking with Joe Weizenbaum at the Center for Advanced Study in Behavioral Sciences in Palo Alto, California.

Okay, let's go back and start with where you went to school and events that led up to your becoming involved in the computer field.

JW: Yes. I think we can skip early history. You perhaps notice I have a little accent. I was born in Germany and came here when I was thirteen years old so that most of my schooling is here. I wound up eventually at Wayne University in Detroit as a graduate student in mathematics and I needed a job. Arvid Jacobson -- if you have him somewhere in your files -- I would be delighted to know whatever happened to him, because I've lost track of him; A. W. Jacobson -- was at Wayne at the time and a fair sized operator. In 1952, I would guess, he managed to get from MIT the Bush Differential Analyzer and also a machine called the Cinema Integrator.

RM: I've never heard of that one.

JW: Well, it is an interesting machine. We might just say a word or two about it. Anyway, of those two machines, the second one was utterly useless. But there it was anyway. Jacobson set up the Bush Differential Analyzer at Wayne and that started a computational laboratory at the university. I became, in effect, a graduate assistant on that thing. So my introduction in computers actually started in analog computation.

Then Jake, as he was known, was feeling ambitious and wanted to get a computer for the university, and he made a deal with Burroughs by way of Harry Huskey who, by the way, came to Wayne as a visiting professor for one year. Burroughs put in a computer that we called UDEC which was a copy of the computer they had in their shop in Philadelphia.

It was made out of standard Burroughs pulse control equipment, it was a drum machine, as I recall it had 800 words of storage altogether and teletype input. A huge machine. A flip-flop was almost exactly the size of a good high-fi receiver today. I recall it had seven tubes in it; a gigantic thing. It required a huge room,

it was forced air cooled, very, very noisy, and very slow. We built that machine there and building meant essentially plugging all those components together, very much like one today plugs together a hi-fi set with cables.

RM: In other words, you bought the componentry from Burroughs?

JW: Exactly what the commercial ins and outs were, I have no idea. Whether we bought the stuff, or whether they lent it to us, or whether they gave it to us, I haven't the slightest idea. In any case, we built the machine which as I say was essentially a copy of a similar machine they had in Philadelphia and Harry Huskey was the leader of that team.

By today's standards it was absolutely incredible. As I said, it was a drum machine, it had 800 words of storage, and the access time to a word on storage was 17 milliseconds. It was a single address machine which meant that there were no tricks you could put to get something like minimum latency access or anything of that kind. You had to wait for a drum revolution to get at every word. The 650, which came

out soon after that as I recall, was a two address machine where you could fiddle around and try to get every word on the drum a little bit more quickly. There was no such facility on our machine. It was a very, very slow machine. It is amazing the sorts of things we tried to do with it when you remember that today 800 words of storage would hardly write a small component of a loading routine or something like that.

As I recall, we could use only the upper case of the teletype so was had very, very few operations codes. For example, I remember that the slash on the machine was the unconditional transfer command and things of that kind. But we had fun with it. We tried a few things. I don't know what ever worked on it and what didn't. I remember at the time an assembly program for it, which was, I think, one of the earliest so-called floating address assemblers. That is to say, you didn't have to specify the location of every word on the drum; there was a process that did that just as today's assemblers do. It was a very, very neat trick in view of the fact that first of all the machine was so slow, and

secondly that the number of symbols we had to play with was extremely small. We had the alphabet, twenty-six letters, then we had a few symbols like slash, plus sign, minus sign, and the numbers, and that is all we had. To write an assembler with that many symbols was quite amazing. We didn't think anything of it at the time. This was the time when in order to work the machine one had to know all about it. Happily, I think, that has disappeared. In general the premium was on being very clever because again the constraints were so enormously tight. Anyway that is one thing we did.

We also had a kind of a first in a way; I suppose it was a first. Occasionally we experimented with using the machine in Philadelphia over telephone lines, so that in a certain sense we had probably the first, or in any case a very early instance of remote access to a machine. It didn't work very well. I remember the sorts of things we had to do. We would get on the teletype and then we would also be on the telephone. There would be somebody on the other end telling us what the light said. [Laughter] It

was a horrible thing. Just before I left I remember a small core memory was added to the machine as a high speed buffer.

Before I get to that though, I think I should add that at one time we had a contest. We got a little differential equation solver going in the machine. Some numerical method like ... or something of that kind. We gave it a problem and at the same time we had a girl on a Monroe desk calculator do exactly the same thing. She won. [Laughter] She was very good, but still that shows something of it.

The other remark that I want to make is with respect to the small core buffer. By the way, I don't know whether you are more interested in people or whether you are interested in technology or ...

RM: Both.

JW: ... both. Okay.

RM: All.

JW: All. Okay. Let me just make this remark which I think will come up again. This is really the first example, but I imagine in our conversation I might site a few other examples. In my view,

anyway, the history of the computer field is full of certain kinds of sinusoidal developments. That is, where things appear and become important and then decrease in importance and disappear altogether. Exactly the opposite takes place for awhile and then the original thing is re-invented, so to speak, or rediscovered, or re-implemented in any case, and it becomes important again and so on. What I have in mind here, is the strange history of a memory which gets another kind of memory added onto it, and then the roles of these two memories begin to interchange. Now, in this particular case, we had a drum machine and we put on a small core buffer. It was a high speed buffer so that main program was on the drum, in fact everything was on the drum, but occasionally for small computations you did work on the small core buffer. Eventually the core buffer gets enlarged and after it reaches a certain size, it becomes the main memory and the drum becomes the auxiliary memory. I think that history has been repeated a number of times.

Just to shoot ahead a little bit, later on when I was with General Electric, a machine was

designed there which I had nothing to do with. It was a machine, as I understood it at the time, for process control, that sort of thing. It was to be a big, huge drum machine, but it didn't turn out very well so they started to add a core buffer. Then somebody had the idea to make the core buffer larger and then eventually the core got to be the main memory and the drum got to be auxiliary. So this repeated. I think we see the culmination of that today, although I am sure it is going to happen again, with these so-called speed large core memories as auxiliary memories to the main memory. We now have high speed core memories as main memories that are a quarter of a million words, with four million word auxiliary core memories backed up by discs and so on and so forth. The idea of the quarter million word main memory as short a time as say eight years ago was just ridiculous. So there is this sort of sinusoidal reoccurrence of phenomenon. There are other instances of that now in the area of memory.

RM: Can you site?

JW: Well, I think we will come to them.

RM: Okay.

JW: Then I left and I took a job with Computer Control Company.

However, before I get to that, just one more historical sidelight. Another machine showed up at Wayne University, just before I left, which I think was a Bendix Digital Differential Analyzer. I don't know who got it. I remember meeting Max Palevsky then. He was a service man and he came out to fix something. [Laughter] That's the first time I met Max and I've known him ever since. We touch base again later. I imagine that he was probably making \$6,000 or \$7,000 a year at the time. Look where he is now.

RM: I call Max Palevsky the American Dream of the Computer Industry.

JW: Yes.

RM: Before we go off Wayne, you talked about the Cinema Integrator and I think it would be fun to talk about it.

JW: Yes. Right. That is a machine that should be somewhere in the history. The Cinema Integrator was apparently an idea of Norbert Wiener's. It was actually built as Gordon Brown's Ph.D. thesis at MIT. It was essentially a heroic exercise

in mechanism control and all that sort of thing. It is one of those things that was probably too early for its time just in terms of the available technology. I imagine it could be done today, although it is probably unnecessary today.

The idea was that various mathematical functions could be represented on 70 mm film. Okay, that is where the cinema comes from. The film was illuminated by a line source of light, which by the way, at that time was terribly hard to achieve and is trivial to achieve today with lasers and things like that, could be achieved only to a fair approximation. The image was then sent into a photo integrator, that is in effect a light meter that measured the intensity of the light that was actually transmitted. This was compared in another photo integrator, I think with the line source itself. I've forgotten the mathematics, but in effect this comparison, which actually resulted in dials being moved by a motor, [Laughter] -- it is extremely primitive -- represented the integral under that portion of the curve that was then being illuminated. The film could actually be moved, sort of like a movie,

and the setting of the indicator could be recorded on paper. So what you got in effect was the integral under the curve that was being illuminated. It was somewhat more complicated than that, but fundamentally that was the idea.

I am sure Gordon Brown got a good Ph.D. thesis out of it, but the machine in effect never worked. It is not clear to me what they thought they would get out of it. It was essentially an analog machine that did integration. Okay. There it was at Wayne. It was never used. I fiddled with it and tried to make it work, but never succeeded. So much for the Cinema Integrator. But I think it deserves a paragraph in history. I imagine that one could find pictures of it at MIT. It would probably be worth looking at Gordon Brown's thesis which is undoubtedly on file in the Electrical Engineering Department at MIT, and it's probably worth mentioning to Gordon Brown. He would certainly remember it. Presumably we will come back to Brown sometime in this discussion, I am sure.

Well, I went to Point Mugu where Lou Fein had set up half his Computer Control Company.

Lou had been in charge of building a machine, the RAYDAC at Raytheon, which was moved to Point Mugu. It was built for the Navy and there it was. Again, this was a very large machine, physically, but a very small machine in terms of power and logic and all that sort of thing. It was a very fast machine by standards at the time. It was electrically very fast, I think its major cycle time was four microseconds or something like that. But it hardly ever worked. [Laughter] I sometimes tell students about it and they just don't believe it.

For example, there was a concept that applied not only to that machine but to all machines called Mean Free Up Time, MFUT or something like that, which today is measured in months, even years. I read the other day that the Dartmouth time sharing system has a down time of something like five minutes a month. In those days we measured it in seconds. A machine would sometimes run for ten, fifteen seconds at a time. That we got anything out of it was just incredible.

RM: It is incredible, isn't it?

JW: Yes, yes. Well, we didn't get much out of it, but we got some things out of it.

The history of that place is itself very interesting. Undoubtedly I am not the one to recite it because I wasn't really in the middle of it and I don't think we should spend too much time on that. But let me just give you some of the highlights as I recall them.

First of all the machine was extraordinarily innovative. It had a lot of things on it that are found today in machines; some things that aren't even found today. For example, it had four tape units. The tapes were optically marked into blocks so you could actually look at them and see black marks. The tape drives had photo sensors that could sense those marks. By the way, the memory was mercury, and each mercury delay line was thirty-two words long. There were thirty-two such delay lines in the main memory; in other words it had 1024 words of memory. Each tape unit had associated with it another thirty-two word mercury memory buffer. What with the tape marking and all that, one could cause a tape unit to search for a particular block for reading purposes, simultaneously cause another tape to search for

a block on another tape for writing purposes and in the meanwhile compute.

RM: It was pretty advanced.

JW: Yes. Well, not only would it search, but in effect you could say, read such and such a block, write such and such a block and compute, and it would do those three things simultaneously.

RM: In other words it wouldn't hold up the machine to do each one.

JW: That is right.

RM: Which was really the state-of-the-art of most machines.

JW: Well it was. Even magnetic tapes were relatively unusual in those days. But the simultaneous reading, writing and computing was simply out of this world. That was unusual.

Of course, if you ordered a tape to be read, you had to be able to tell whether the reading had actually taken place or not, because it might take ten minutes or it might be in the next few seconds depending on how far away the block was when you initiated the operation.

By the way it was a four address machine. There was a command test reading completion. You

could cycle on that. You could test it and then branch back it. It was essentially a branch command. When that flip-flop turned over then you would go on. So you could loop on that. Similarly for the teletype, which was connected to the machine. You could put things in directly through the teletype, and there was also a command that would test the teletype to see whether the writing operation had been completed. You could issue a command 'read from teletype' and then it would wait for the carriage return. When that was finished that would set a flip-flop saying in effect, "Okay, I have read from the teletype."

I wrote a very small program that would fit into thirty-two words. In other words you could put it on one block of tape. Then I told the maintenance engineers, "Okay, now clear the machine, mount the tape, and you will see output, a series of numbers coming out of the typewriter. Now stop the machine, rewind the tape, dismount the tape, clear the machine and then go through the same process again. The next time it will produce an entirely different set of numbers."

I asked them "Either this is a determined machine, or it isn't. What is going on? There must be something wrong." What I had done, and it took them a hell of a time to find it, was that in that thirty-two word program on the tape was written a little program which called for input from the typewriter and then cycled on waiting for the typewriter. While it was cycling I would perform some additions. Then when the typewriter said, "Okay, I have read" -- you have to type in something -- I would take the resulting number that I had counted out and do some tricks with it, like raise it to the seventh power or take the middle fifteen bits and print those out. Because the typewriter is a mechanical device the time it would take each time it did that was variable. Essentially I was counting how fast the typewriter operated. It was very sloppy. Then the typewriter would type out another number I would again cycle on how long it took the typewriter to type and produce another, in effect, pseudo random number and type that out, and so on and so forth. Of course, everytime you did it it came out differently. Anyway, that was a little trick I did with that machine.

Another innovation. I mentioned earlier the business of using that machine in Philadelphia long distance. Another sort of a first on the RAYDAC was a scope mounted on the console, and being a delay line machine you could switch the scope to look at any one word in the main memory, in fact in any part of the memory main or auxiliary. What you would see was the actual pulse pattern. I wrote a differential equation solver for that machine and one of the things we used to do was turn the scope toward that word in memory where the error term was being maintained, and by watching it you could see it get larger and smaller. If it got beyond a certain threshold, you could throw a switch which would have the interval of integration of the process. When it got below a certain threshold you could throw another switch to double it. So in effect we had the first sort of man-machine interaction; you know, dynamic intervention problems. I really shouldn't say first, but at least an early instance. There might very well have been others then. Of course the Bush Differential Analyzer was a machine that was used in a

man-machine interactive mode. I remember sitting at the tables, the big tables where you would do curve following by hand, and that would be an input to the differential analyzer. That was a man-machine interaction, but it was really a different kind.

RM: This was man-machine interaction with a fully electronic computer.

JW: Yes, exactly. And digital and so on. Interactive computing was something that was to show up again later.

I think one way to characterize my own involvement in the computing field is as a series of disillusionments. That would be a very incomplete way of characterizing it. It is not that I have gone through the field constantly being disillusioned, but one kind of marker that keeps showing up is a kind of disillusionment. We will come back to that I am sure. So that is a sort of sub-theme that runs throughout. There were other themes and I want to emphasize that. Perhaps my first one was when I discovered, and this is just a personal matter, elementary logic. It became clear to me that with elementary logic

you could do anything. Pretty soon I discovered that perhaps you could do anything in principle, whatever that means, but it could get very complicated very soon and that is not a really practical idea.

At Point Mugu we were running a Computer Center fundamentally for the Navy at that particular installation. I guess my main contribution to it was this differential equation solver that I wrote and actually got debugged. The RAYDAD was an incredible machine, it would run minutes at a time, at best. People would come in with sets of differential equations to be solved and it became very clear to me that they hadn't the faintest idea what they were doing. I think we had two or three differential equation solvers and which one you chose to use on a particular set of differential equations should have depended on the mathematical characteristics of the equations, on their stability and so on and so forth. The engineers didn't know anything about that, and they would come in and say, "Well, I don't know, just solve it. I just want a solution." I got the intuition that there was something terribly

wrong, and I think that intuition has been reinforced for me over the years, many, many times in sometimes much more serious ways. In other words, people use computers as black boxes. They don't know what the hell is going on inside, but they really don't understand their problems. All they want is for the computer to say something and whatever the computer "says", that is what they will accept and that is what they base their decisions on.

RM: A lazy man's way of solving problems.

JW: Yes. I think that this particular phenomenon has actually little to do with the computer per se. It is a very wide spread phenomenon which in ordinary life, not having to do with computers, manifests itself mainly in reliance on the expert. You go to the expert and really all you want is an answer. You don't understand your problem and he tries to explain your problem to you or get it out of you. That is not what you want; you just want him to do something, say something, whatever. I think in a way, and I don't think it is a very large stretch of the imagination at all, one can account for a very large part of

the various unfortunate things that have happened in our recent history in the United States, Viet Nam for example, the Watergate business is another example, the IT&T business is yet another example. One can go back and find many other examples where at least from a certain point of view the fundamental difficulty is essentially that no one really took the responsibility of trying to discover what the problem was. Everyone took his assignment to be the application of his particular technique to the problem as posed, without looking back to see what the problem really was.

This is what happened at Point Mugu and it became very visible to me. People would come in with differential equations. They didn't want to understand what they were doing, they just wanted a solution. They wanted the big machine or the big mathematical experts or whoever to come in and produce an answer which they would then use. As I say, I think it is analogous to a bunch of political scientists being asked what to do about making villages safe in Vietnam. They take that to be a technical task without

asking other questions such as, "What is our context? What does this problem really mean?" Is it appropriate to ask the kinds of technical questions we are trained to ask about this particular problem?

RM: It's a very close analogy. Were you aware at the time we are talking about at Point Mugu of social implications of the computer?

JW: No, I don't think so. As I say, I was troubled by this phenomenon that I witnessed, but I can't say that I in any sense integrated it into larger areas of thought that I might have been engaged in. No, I don't think so. It just troubled me. I thought it was terrible. I didn't really think much more about it than that.

Eventually Point Mugu got a 650, which effectively dominated the RAYDAC. There was nothing the RAYDAC could do that the 650 couldn't do better. I guess eventually the Navy decided to retire it, but before it did I had already left.

Now a little bit of history. I take it you are going to talk to Fein?

RM: Fein has been talked to.

JW: Oh, he has been talked to. Okay, so you will get other aspects. I was just an employee, of course, but I got very friendly with him, his family, his children and so on. I still know them. At the time there was a small revolution in his company. He was president and there was a branch of it back East. He was, I suppose deposed is the word. One thing I learned from that experience is how difficult it is to know anything even about current history. I was friendly with Lou, I was friendly with the other people who were perhaps on the other side, and I could never find out what happened. Not that anyone was being secretive. It was just that essentially there isn't any one reality. It isn't even that there are three stories and the truth is "somewhere in between." It isn't like that either. There just isn't any single reality; there are lots of realities. That impressed me very much at the time. I felt how difficult it must be to be a historian when you are not there at all. I was there, I knew all the people, I was friendly with them, they were all willing to speak with me, and still I couldn't really

come up with a coherent picture of what happened. Not that I tried very hard. Anyway, if for nothing else but to synchronize the various stories you hear, that was another event that took place at that time.

RM: Yes. Just let me get this story a little bit straighter. Raytheon built RAYDAC, is that right?

JW: Right.

RM: And Raytheon was acquired by Honeywell?

JW: That was much later. Raytheon built the RAYDAC and apparently Lou was the project manager for it. After it got more or less finished, Lou apparently decided to form a company to bid for the contract of actually running the RAYDAC for the Navy. He won that contract and so he established the company and among other things they built pulse control equipment and all that sort of thing. That was CCC which had a history of its own afterwards that I have only hear-say knowledge about. I don't think I can contribute anything to that. Raytheon, some time later, rid itself of its computer business and, in effect, promised not to re-engage in it for sometime as

part of the deal. There was an outfit in Boston, Datamatic, that sound right, that was building machines and I think the rest of the Raytheon gang went there. That eventually became Honeywell Computer Division, and then of course, the whole thing merged once more. Honeywell bought up CCC, I guess, so the whole thing was a big homecoming. From my own personal point of view, an interesting thing is that eventually Honeywell bought up the General Electric Computer business. We will come to that. So there is a bit of corporate history but I'm not a very good person to ask.

RM: The other thing I need is the approximate date when you joined Computer Control Company.

JW: Well, it must have been something like 1953. I don't actually remember, but I would say 1953. I was there about two years, and that is really where I got wholly involved in digital computing and programming and so on as a profession. There were a number of people there who I remember fairly fondly. One of them is Bob Waller, who is now at III, Information International Incorporated, and John Hanson, who is also with III. They were two very, very good people, extremely

competent and I was very impressed with them at the time.

RM: What was their field? Programming?

JW: Hanson was a programmer and still is. Bob Waller is sort of an all-around genius engineer. He is very, very good. I don't know whether you should talk with him or not, but he is a very good guy. Knows a lot, very strong and so on.

At the time, I must say, the computing business was sort of a journeyman business. People just went from one project to another, it was quite common at the time. When I decided to leave Computer Control Corporation, this is after the revolution, I really didn't quite know where I would go. There were very good programmers around and I knew I would get a job, so there was no problem with that. I applied to SRI, who didn't take me, and I think of that fondly every once in awhile. I wound up once more in a way with Harry Huskey, because I got a full-time consulting contract with the Bendix Computer Division.

My boss there was Dave Evans, whom I much admired. I've had some pretty good luck. Dave

Evans was a very, very good guy; Harry Huskey is a good guy. I've worked with these two people and others and I think I'm lucky to have known them. I value honesty, or perhaps I mean intellectual integrity, or something like that very highly. I think both Huskey and Dave Evans exemplified these qualities to a very high degree.

RM: Okay. Let's switch sides before we go on.

END OF SIDE I

SIDE II

JW: Bendix had the G-15, a small computer that Harry Huskey had a vary large hand in designing.

I understand, by the way, that there was some financial dealings with respect to patents and rights and so on, that apparently made Harry Huskey not unreasonably wealthy and hurt the feelings of lots of other people. Generally I am niave about these things and essentially I am just reporting what I have heard. I have no idea what actually happened, all I know is that Harry managed to live very well. I imagine this may have something to do with the G-15 and his rights to it.

In any case, they had the machine. In a way I suppose one might say that the G-15 was a kind of forerunner of the Hewlett-Packard pocket computer of today. It was about as small a machine as one could build at the time with the technology. It was designed to be a mass computer, I guess, and its direct competitor was the LGP-30 which was designed by Stan Frankel whom I dearly hope you have talked to, or will talk to.

RM: Absolutely have. Lovely man.

JW: Lovely, lovely man.

Anyway, I think the basic problem with the G-15 was that it was designed by engineers who really had no idea of what a programmer does. While it was enormously clever and very, very elegant from an engineering point of view, it was terribly, terribly hard to program.

RM: What made it hard to program?

JW: The programming was essentially what we today call micro-programming. That is the way one controlled what that machine did; by almost literally commanding the opening and closing of gates and so on. One had to have a rather intimate knowledge of at least the register structure of the machine, and the information flow through the machine, in order to be able to write a reasonable program for it. Well, not even a reasonable program, even an unreasonable program. It was really enormously difficult to program and given the market to which it was presumably addressed it was almost a contradiction in terms. Because while you might suppose that a very large machine, say a million, two million dollar machine would be administered by a staff of highly trained

professionals, this little tiny machine which was designed to go into small universities, high schools, small businesses and so forth ought not to have been conceived with that as a prerequisite. That was essentially the difficulty.

I was hired for a year, to work up an interpretive system for the machine. It was a programming system called INTERCOM and the aim was to make it easy for the ordinary guy to get at it and to program the thing. The machine I was given access to was actually in Harry Huskey's home in Berkeley. The story was told at the time and it never died so I am going to try to kill it once more, but it won't die anyway, that it was in his bedroom. This was the first example of the machine in the bedroom myths which haunt us today. Today at MIT lots of people have consoles in their homes. In fact, today in lots of places, people have consoles in their homes. The story is always that the console is in the bedroom. As far as I know, they almost never are. The story was that the machine was in Huskey's bedroom. It wasn't.

In any case, I lived in Oakland and worked in Berkeley on the machine. With Huskey's help,

and I must acknowledge that working on that system under Harry's guidance was to me an important part of my education, I did the actual programming and coding for it, but many of the very good ideas were Harry's and I learned a lot. There was also a fellow at Humble Oil Company in Houston. They had a machine and he too helped out. I've forgotten his name. I remember going down there occasionally. A very good guy; a very competent guy. In a way one must really say that the INTERCOM was a three-man effort. There was Harry, this fellow at Houston and myself who worked this thing out.

I left after a year when essentially the job was done.

RM: INTERCOM was the package then that went out with the machine?

JW: Exactly, and I think it did its job. The machine was sold fundamentally as an INTERCOM machine. The general user never actually saw the G-15 anymore; he saw the machine through the INTERCOM interface. It was a kind of an interpretive operative system.

RM: So you removed the need for getting in and understanding the register layout?

JW: Yes. Oh, yes. Absolutely. The job was finished and so again this journeyman business; I left. Later on this fellow in Houston, I don't know what Harry's involvement was in it after that, and other people came up with INTERCOM II, INTERCOM III and it got fancier and fancier. I was not responsible for the later developments which I understood were really very good and really made the machine very, very usable.

RM: This would have been for later Bendix machines?

JW: No, no. It was the G-15.

While I was still with Bendix I used to go down to Los Angeles to meet with people down there about once a month or perhaps more often. I would consult with Dave Evans and we connected with Max Palevsky who was working there at the time. They were trying to think about a new machine which ultimately turned out to be the G-20. That is where I first tasted Max's ambition. He was much interested in becoming independent, not to say making a lot of money. I think he left Bendix to go to Packard Bell while I was still with them, but I'm not sure about the exact time sequence. Anyway I reconnected with Max there.

RM: Tell me a little bit about Dave Evans and what his contribution was.

JW: He was Manager of Engineering at Bendix Computer Division. I knew him essentially as an administrator, and I was impressed with his honesty, openness and all that sort of thing, and his intellectual qualities. He seemed to be a very, very smart guy. I could always go to him with either technical or administrative problems and always get help from him one way or another. I just thoroughly liked him. I'm not sure I can say much more.

After I left Bendix I went to General Electric, and I'll say more about that in a moment, but sometime later, just to shoot ahead just a little bit, Dave did show me some sketches of the G-20, which I thought to be a very clever machine. It was a forerunner, I think, in many ways to modern machines. It consisted of several machines communicating with one another and so on. It almost anticipated in a way the ARPA network except that everything was in one room. I thought it was a very clever machine. I really don't know why it didn't fly. I would suspect that it

had to do with entirely non-technical matters; financing, marketing, difficulty in coping with IBM, and so on. But the machine impressed me at the time as being a rather good, clever, clever isn't quite the word for it, forward-looking machine.

I'm still in the journeyman phase of things and I knew that my relationship with Bendix was coming to an end. I seem to recall getting an offer from Bendix to move down and join their operation but I didn't feel like moving to Los Angeles. I started looking around as to what I might do. I remember at the time an IBM ad appearing in the newspapers asking, "Are you interested in language? Do you speak a foreign language? Are you interested in playing chess?" In effect what they did was to draw a profile of me. "Do you have several years experience with computers?" which, of course, at the time was still relatively unusual. "Do you know math?" So in effect they had a profile of me. "If so, we have a wonderful job for you." I guess at Yorktown or Poughkeepsie, or someplace like that. I very seriously thought of going with IBM at the time.

Then I heard somehow that General Electric was bidding on a Bank of America contract for the ERMA machine. How I heard this, I don't know. It is even conceivable that I saw an ad in the paper, I just don't know. I really knocked on their door just off the street and got interviewed in San Francisco by a Mr. Green. I got hired. The place was here in Palo Alto. So my involvement with ERMA began at that time.

RM: So you came in just as G.E. got the contract?

JW: Actually, yes, that is right substantially, although I don't think the contract was signed until after the last machine was delivered.

[Laughter] But yes, I came in very nearly at the beginning. The people were here at Palo Alto, actually Menlo Park at SRI and the man in charge was Bob Johnson. I surely hope you'll interview him.

RM: Yes. These are things I have needed because I haven't gotten them yet.

JW: Oh, well, you must absolutely talk to Robert R. Johnson. He is now Vice-President for Engineering at Burroughs Computer Division in Detroit. He is a considerable big shot. Most of the people I

worked with have made out pretty well. You absolutely must talk with him. He was in charge of ERMA for General Electric. He was hired by General Electric -- to head the team to do the ERMA job. So he was the boss. A fairly young man, quite bright, easy going type. You will enjoy meeting him. He knows everything about ERMA from a particular point of view. Later on he became Manager of Engineering for General Electric, so he knows a lot of politics and other things like that. You surely cannot get a complete ERMA story, or for that matter a General Electric story, you know what happened to General Electric, without talking to Bob. He was the boss, there were three other people who were important, and then a few more junior people. But the team was just being assembled. The three other people were John Pavenon, do you have him down to talk to?

RM: Yes.

JW: Henry Herold, the logic designer who later went with Palevsky, and Jay Leventhal. Pavenon was hardware. Herold, logic design, and Jay Leventhal, sort of over all systems design. Then I came in

to do, essentially, software. So it is Bob Johnson and the four of us that were trying to do this thing. I think all of them had sort of walked in off the street to do the job. It was a really excellent group.

Now let's get to ERMA. The first thing I was told was that SRI had built a machine to do the ERMA job and I went to see it. It was an incredible machine. It was just enormous. Good God! I have no idea how many cubic feet of space it took. It was an enormous machine. A big warehouse full of gadgets. It was a wire program machine. It was just an insanity, an absolute insanity. Everytime you wanted to change something, you had to go in with a soldering iron. Debugging is hard enough as it is, but this was simply incredible.

RM: Did they ever tell you why they went this route?

JW: I think they just didn't know any better. They had the idea that a bank is a bank is a bank, and that deposit accounting is deposit accounting is deposit accounting, and that if you did the job once you have done it forever. Even when I would do it with a stored program computer,

their idea was that since it had to be done only once, you hard-wire it and that is the end of it. They were incredibly naive and built this huge monster, and apparently Bank of America said to the world that they wanted this particular machine product engineered; they wanted a machine just like it except product engineered, whatever that means; it could be mass produced or whatever.

Now, of course, you have to understand that I was an enormous little shot in G.E. so all this high level stuff I got by osmosis and rumor mill and so on. I can't speak very authoritatively about that, but Bob Johnson can. He will surely tell you lots of things. Apparently General Electric had decided that it wanted to be in business, and that they wanted one big contract to sort of establish them. They realized that they might very well lose a lot of money on it, but nevertheless a contract of this magnitude would help them get started in the business. So they went after it.

Other people, I was told, had looked at the job and effectively decided that either it

couldn't be done or that it was an insanity or in any case they didn't want to do it. Apparently IBM had given the Bank of America its usual solution; namely, we will sell you a machine that will do the job and build the system for you, and we don't even want to look at what SRI did. The bank had a lot invested in SRI, I don't know if it was large financially, but certainly it was large emotionally and all that, and they wanted that machine.

When we first looked at it we thought that the best thing to do even then was to go out and buy some existing machine. For example, there was a machine that was fairly well known in those days called the DATATRON. We could have built that system on the DATATRON for probably less money than for almost anything else. In other words do what today would be called simply a systems engineering job where you buy this machine, that drum, that software, those tapes and so on and so forth and then although you still have to do a lot of software, you get the system out of essentially off-the-shelf hardware. The bank didn't want to hear anything about that.

How we managed to get out from under having to build the SRI machine, I have no idea. Maybe we just did it and just told them we were product engineering the SRI machine when in fact we weren't doing any such thing. What we did is to design a new machine; the ERMA machine. It had some other name G.E. too, but I can't remember. Pavenon with Henry Herold fundamentally designed the machine. I got to know Stan Frankel because he became a consultant to the project.

The first silliness in the machine is that it had to be an internally decimal machine, because after all, this was for the bank. That gives you some clue as to the degree of sophistication of our management. They didn't realize that the decimal is ultimately binary at some level, but no, it had to be a decimal machine so that bankers could understand it. That turned out to be an omen of things to come, I think.

I don't really know what to say about it. We worked very hard trying to understand the banking system. Jay Leventhal did most of the real systems work trying to understand just what banking was all about.

RM: What was your relationship with SRI?

JW: Well, there were some people who were closely connected with SRI because of the character reader check handler, but I didn't have anything to do with that. A guy named Dave Zeaheb, Z-e-a-h-e-b, I believe, did most of that stuff, although I am not sure. Again, Bob Johnson would be able to tell you a lot about that. Certainly my gang doing the actual programming for the machine, had relatively little connection with SRI. I think we just went ahead and did our job.

RM: What were some of the problems that you had to overcome to get this thing working?

JW: The first year and a half I don't know whether it was a year and a half, but for some very long time the problem was one of getting the machine going. Just building the machine, the logic design and all that sort of thing. Fundamentally, it was a general purpose machine, digital computer. There were decisions we had to make about word size, size of memory and all that sort of thing which involved some estimate of what the job would turn out to be, the number

of tape units and all that sort of thing. But strictly from my point of view, again you have to remember I was at a fairly low level, it was a job of designing a machine, designing languages for the machine and we wanted to simulate the machine. At the time, General Electric's Computer Department was actually in Phoenix where they had a 704. That is how I got to know Dan McCracken and Herb Grosch. We simulated the logic of the machine on the 704. I was deeply involved in the banking aspects of it. Jay Leventhal did most of the guidance on that. My problems were pretty much technical problems, working out sorting techniques because the tapes had to be sorted, all that in the light of what the bank job would turn out to be.

That was a very long time ago and I haven't thought a lot about it, but my current impression is that my group had almost no contact with SRI and relatively little contact with the bank. That information as to how the banking aspect of it was to work came to me mainly through Jay Leventhal.

There were a lot of difficult technical problems; integrating the check handling business with the magnetic ink and that sort of thing into the software, lots of timing problems and things of that kind. That is what I remember. It may be that I'm suffering a little bit from some sort of merciful amnesia because there were some very traumatic times that I have just simply put out of mind. That is quite possible. In fact, the whole experience when we were working very hard on the machine is just kind of a blur. We just worked incredibly hard as is quite usual on big projects that have a very definite goal, and a bunch of excited, turned-on people working on it. Work and work and work and work, and eventually when you get done the whole thing just looks like one project and all the details are forgotten. It is sort of like memories of being in the Army, [laughter] eventually you just remember that you were in the Army. I am really at a loss as to what to say about it.

I remember there were certainly crunches when one deadline would pass and another one and

then questions arose as to whether it could be done at all. People would come down from New York and there would be agonizing decision making. Bob Johnson would be in a much better position to tell you about this. I am sure there was at least one time where General Electric really had to decide to either give up and pay the penalty, whatever it was, or just make the thing open ended -- whatever it cost, and finish it. The latter decision was made and we continued.

In the meanwhile in Phoenix, back at the ranch, another operation had started. The real computer department was there. For some reason that I can't divine at all, the company made the decision to go to bed with National Cash Register; they built the National Cash Register machine there. Apparently it had been designed by NCR and was being manufactured by General Electric for marketing by General Electric. I think General Electric had the right, and indeed the obligation, to market within General Electric, and NCR could market it outside. It was a terrible mistake. It was just a terrible mistake. It was a horrible machine. It was another

instance, and there were so many in those days, of machines being designed by electrical engineers who simply had no idea of what machines ought to be used for, how they would be used, programming, or anything like that.

Quite naturally a kind of struggle developed between the Palo Alto group and the Phoenix group. We thought we had a pretty good machine, and we did. And they, of course, thought they had a pretty good machine. Their machine was unique in that it had a micro programming control unit. Do you know what that is?

RM: No.

JW: Let me try to give you a two minute lecture on that. When I talked about the G-15, I said effectively the way this thing was programmed was that in a command certain positions were one and certain positions were zero. We actually had to know that if this bit was a one, then that opened certain gates and so on, so you were fundamentally controlling the machine at a micro level.

RM: Yes.

JW: Okay. Now most machines weren't like that. The NCR wasn't like that either. If you think of

the command which contains all sorts of bit patterns, that eventually goes to some command decoder which then takes it apart and its job is to cause certain gates to open and all that sort of thing. With the NCR machine, that decoder itself was something that was capable of being modified. So that a command that might mean one thing in configuration of this little modifiable command decoder, would mean something altogether different if the modifiable command decoder was in configuration B. It was subject to being modified in ultimate detail. It was itself a kind of big switching network that was capable of taking on an enormously large number of states, all this by wiring. It was essentially a kind of a core memory through which one could filter information and when it filtered through it was modified by the filtering process. That was one of its big features. Much later that same idea reoccurred, this is another one of these sinusoidal things that I mentioned. The earliest computers were indeed microprogrammed to begin with. The G-15 was moderately microprogrammed. Later machines weren't microprogrammed

at all. Then we have this microprogram control unit in the NCR machine, the name of which I have forgotten.

RM: 304?

JW: 304, that is right. The NCR 304 or the GE 304, whatever it was called. Then much later people got the idea, which by the way is not a bad idea, to take microprogram control units, MPCU's sometimes called, and to make them reconfigurable by program as opposed to actually going in and taking out cores and putting in plastic cores. Things of that kind. The significance of a plastic core, by the way, is that it is always zero, it is not changeable at all. People got the idea of making these things themselves, programmable, so now we could refigure a machine entirely by program, which is sort of equivalent to picking yourself up by your hair and holding onto your head at arm's length, or something of that kind. That was a moderately good idea under certain circumstances for certain applications. Today, the emulators we hear of are the direct descendents of that concept.

That was one of the big things on that machine, but it turned out to be absolutely impractical mainly because the changes had to be made by hand. You have to ship the units somewhere, they have to be debugged all over again and that sort of thing, and nobody wanted to buy a machine, I think, that turned out to be reconfigured every few hours or something of that kind. You would start writing languages for the machine and pretty soon you would begin to depend on the machine being what you knew it to be. What was somebody's very bright idea ...

RM: A little too early?

JW: Well, too early and not a good idea in that form. It is certainly not a good idea for a general purpose or commercial computer, but again for very special purposes it might be a good thing.

RM: It is interesting because the group of people who started that design was the CRC group, and they were really very special purpose computer minded. DDA's were their background.

JW: Yes, right. And the thing that you are putting your finger on to some extent is what I also have tried to say when I talk about engineers designing

computers. They really looked at it as a series of difficult engineering problems for which they had very clever solutions, but absolutely no appreciation for the fact that such things need to be maintained in many different senses, and that there are going to be quite ordinary people out in that big world who are going to try to use that machine for quite ordinary problems, like inventory or whatever. It was a disaster. The machine was late.

The team was assembled at General Electric in Phoenix. I think another thing that ultimately in some sense helped to predetermine the demise of the outfit is that General Electric had, and perhaps has to this day, a Navy concept of management. First of all that management is a science and that people learn to manage independent of what it is they are managing. That is one thing. Another thing is that certain jobs require management of a certain level. Okay, a job in the Navy might call for a Lieutenant Commander. It doesn't matter whether he is an aviator or submarine guy or what he is, he is a Lieutenant Commander and that means he can do

that job. Well, General Electric has the idea that a manager is a manager is a manager, and when the Computer Department needed managers of various types, a call went out, especially in G.E. land in Schenectady and Evendale and places like that, saying we need a guy like this Lieutenant Commander, so to speak.

Of course, the people who were being creative at the time and who were happy and working on something they thought important wouldn't apply for that job. Whereas all the guys who had had it and thought this was the chance to get out and possibly get a new start, or goof off in the sun in Arizona or whatever, are the ones who applied. So the middle and top management in Phoenix was largely a bunch of guys who had had it somewhere else and who certainly didn't know anything about computers. One of the things they didn't know, and this is pretty important, is they didn't know how to judge whether an applicant coming in knew anything about computers. In a certain sense they were just terribly easily snowed by some guy who had put in two years with IBM or RCA or wherever, and was leaving because

he wasn't successful. So it was quite an incompetent team in Phoenix, in my view anyway, and they just got into a hell of a lot of trouble.

They also decided that they needed a process control computer and this reflected a very high-level management decision made all the way up at 570 Lexington Avenue, New York to the effect that General Electric could never compete with IBM on the broad front, and General Electric had to do the sorts of things that it was especially competent to do and where they had special ins with the market. Okay, one of these was process control. General Electric had the concept of the turn-key job, meaning that somebody would order a factory or a power generation station and go to General Electric. General Electric at some later time would hand the guy a key to the complete plant, which he then would turn on and it would all be there. Presumably computers were needed to help turn on and monitor power generation stations, nuclear and otherwise, so process control was a big thing.

Again, the utter naivete of the company with respect to computers did them in. Process control

is a very, very difficult field, there is an enormous amount of theory that has to be understood, a field known as control theory. What General Electric did was to hire some electrical engineers who at some time or other had designed a computer somewhere and just knew nothing. Anyway a process control computer got started under a guy named Arnold Spielberg, who had come from RCA. You also have to appreciate that the people who were brought in became terribly emotionally involved. Their careers were at stake and it was a pretty heavy thing for everybody. Well, the process control computer fell flat on its face.

RM: Did it ever get to production?

JW: One was installed somewhere in southern California to help run the power station down there, and all I remember hearing about it is that it was just an incredible disaster. It may even be that that computer was pulled out and General Electric just bought someone else's computer and put it in. It was that bad. It was a terrible machine.

In order to rescue it, they turned that computer into what later became the 225 of the G.E.

product line. As I indicated very early, it started out to be a drum computer, then it had a little core buffer, and then it eventually became a core computer with little drum buffer, and so on. An incredible piece of design. When you think of a major company designing something you would think it would be thought through and so on. No such thing.

RM: No only that. It sounds like it took years to work through with this design.

JW: Oh, yes. Years and years and years. Things that today can be done in a couple of months took years and years and years of work. It was very, very sloppy.

RM: Did they finally market this?

JW: They did finally market the 225 with very poor success. I doubt if very many people remember it today. It was just not a very good computer. That, of course, also had the effect of inducing good people to leave. People who had feelings saw the incredible sloppiness of this thing and the amateurish way in which it was done. They just left. People like Dan McCracken, for

example, could have helped the place survive.
But it was very discouraging.

END OF SIDE II



TAPE II, SIDE I

RM: When you built the ERMA machine I take it you were in a position to work close enough with the engineers so this was an easily programmable machine?

JW: In Palo Alto we worked intimately with one another on every aspect of the design so that hardware changes and even original designs were induced by programming considerations. For example, I remember that we put in an instruction called Tumble, which was supposed to help in sorting, and I think it did. We discussed the software requirement emanating from the system requirement with Henry Herold, and he designed the logic to do this tumbling thing. So we had a Tumble instruction in the machine as dictated by this other requirement. The interaction between the engineers, the logic designers and programmers, was very, very intimate.

RM: What is intriguing is that General Electric didn't take what they had learned from ERMA and apply it to the later machines.

JW: Much that I'm talking about happened in Phoenix simultaneously with the ERMA design, so they were

off on their own kick in those days. Our attitude here in Palo Alto, and you have to understand we weren't entirely neutral, was that the ERMA machine was in fact a good general purpose machine. At the very least for business purposes, but quite generally. It was just a general purpose machine. The only thing that distinguished it from some other machines is that it was internally decimal. It was just a very fine machine for its time. It was the first solid state machine manufactured anywhere -- reasonably fast and so on.

RM: Maintainable?

JW: Maintenance is the only thing we worried about, the bank would have to maintain it. And eminantly programmable and so on. And we thought General Electric should market it as its project. And those chose instead of market the 304 including inside General Electric. It was just a terrible mistake.

And the other thing, and I hope you get some insight on this from the IBM side from those days, was that after a couple of years or whatever it took a few years, I have forgotten, we

had a running banking system and IBM had nothing. I have been told by people who were in IBM at the time that IBM experienced one of its rare moments of being frightened. Here was real competition in this area. Here was real competition in that area and what General Electric should have done at that point; especially with the help that Al Zipp was volunteering -- shoving down our throats.

He was going around the country extolling this particular concept and saying that we had delivered a good machine. We should have mounted a major campaign to get G.E. computers into every major bank in the United States. You can go back in time and look at the Wall Street Journal and the New York Times for example, and see the ads that we placed for our machine. Eighth of a page ads. Like advertising used cars. There was absolutely no concentrated advertising campaign or anything of that kind. There was simply a failure on the part of management to understand what they in fact had. I think that contributed mightily to crashing the whole outfit. Somehow, somebody at a very high level

had gone to bed with the President of NCR and there was this wedding between NCR and General Electric that practically hypnotized higher level management. NCR being a very big, important established old American company, and all that sort of thing.

G.E. just didn't understand computing. Nor did they understand management in a field that involved rapidly changing technology. Now I think their management concept of the professional management, the manager who manages anything can manage anything else, probably applied to areas where the technology is very stable. Steam turbines, light bulbs, small aircraft engines, or whatever. I don't know how stable small aircraft engines are! The computer field was changing enormously rapidly at the time, as it still is, and it was necessary, as it still is I believe, that high-level management and management at all other levels be capable of keeping up with the technological changes that are constantly going on. I think our original General Manager, apart from the guy who just got it going, was a guy named Claire Lasher,

and he just hadn't any idea what was going on. The Manager of Engineering, a guy named Ken Geiser, is perhaps a good old fashioned engineer but he didn't know anything about electronics and he certainly didn't know anything about computers. Nobody knew anything about software or what software would cost, for example. We hired a guy named Charlie Katz, who became the Manager of Software for the Computer Department. He was a absolute amateur. He had been in the field for a long time, or for what counted for a long time in those days, and he was sort of like one of the senators who gets where he is by having sat on his hands for a long time and managed to survive somehow. It was a terribly disorganized software for; no one knew how to run the thing, or how to make estimates or what needed to be done next, and so on and so forth. Product Planning was another disaster area. A guy named Lou Nolfry, he later on went to UNIVAC and I think got to be a fair big shot, was Manager of Product Planning for computer manufacturing, and he just didn't know anything. It was really an absolutely amateur effort.

RM: That is where IBM was strong.

JW: Yes. They were just a bunch of amateurs.

RM: Then, of course, IBM actually started to educate the executives of their customers!

JW: We just had nothing like it. But we could have. I think we could have by concentrating on a small area and getting some really expert help. For example, if someone had realized that process control really involved two or three people who understand control theory and you can't just do it by sort of bolting your way through, we might have done something there.

RM: There again is a case of desperately needing to understand the problem before you start to work on it.

JW: Exactly. That is exactly why that failure cropped up over and over again and I think was the failure to market the ERMA machine either in banks or in the company itself. After all, the company had an enormous bookkeeping problem. Even as a general purpose computer to do scientific computing and so on, the machine was a perfectly suitable machine. G.E. was paying enormous amounts of money to IBM at the time. I remember

hearing the figure of twelve million dollars rental that we were paying one of those years to IBM. If we could have replaced half those computers by G.E. computers, that would have been a big market right inside the company. Instead, the word got out that the NCR 304 was to be marketed within General Electric, and nobody wanted it. Individual departments within General Electric had their own profit and loss responsibility; they are encouraged to buy G.E., but they don't have to, and they didn't.

I think there are two more things to say before I take my leave from General Electric. I happened to be in Phoenix one day and I saw a brochure describing a computer on somebody's desk. It was not possible to tell from the brochure who's computer this was. I looked at it, I looked at the specifications and it looked good to me. "Who's computer is that?" "Shhhhh, don't tell anybody, it is a machine that was designed for the military in Schenectady."

By virtue of some internal company agreement, they weren't supposed to do that; machines will be designed by the Computer Department. But they

did it anyway, and they had sold it to the military somewhere. I've forgotten the name of the guy who designed it. I thought it was an excellent machine based on the paper description, and being generally fearless because I was not ambitious, that characterized my career throughout, I said, "The hell with it, if it is a General Electric machine and here we are in General Electric, I don't care if it was bootleg, somebody should know about it."

So I went up and down the corridors, talked to the managers and went to New York and told Mr. Strickland about it. Well, everybody was very unhappy because of the political situation. These guys shouldn't have done it, it was a bootleg thing. But damn it, we had a good machine and in-house in a sense. It was designed to military specs, which meant it was much too expensive for commercial purposes. It had to be radiation resistant and built so it could be dropped from an airplane, God knows what. In fact, with the last breaths of the Computer Department that's the machine that ultimately became the G.E. 235. That ultimately

became the 600 series, and then became the time sharing machine at MIT. You know, here is an evolution of that design. That fellow, whatever his name was, was brought into the Computer Department and began to make some sense. I guess by that time the rot had set in and too much money had been spent, and management just got tired of supporting the effort.

Meanwhile back here we finished the job and we had a bunch of people who had obviously done a good job and knew how to work with each other and were reasonably bright and all that sort of thing. In the meanwhile the laboratory got moved to Sunnyvale and neither in New York or in Phoenix management knew what the hell to do with us. It was a question of indecision. What they should have done probably at that point is to say, "All right, you guys move to Phoenix, and those of you who don't want to, well, goodbye." But they just couldn't bring themselves to decide that and the thing just drifted along. It took three or four years to peter out.

RM: What were you doing?

JW: Well, it was then labeled a research laboratory. We started a number of "research projects," and one of the things we looked at was cryogenics. While that turned out to be a bad idea, it was the proper thing to do at the time. The next thing that came along, although that couldn't have been known at the time, was micro-miniaturation, chips, large scale integration, and all that stuff. So we worked on cryogenics for a while. I didn't, but some people at the lab did.

As for me, personally, nobody told me what to do and there really wasn't anything to do, so I started to work on a list processor. God knows how I got that idea. But I finished SLIP, a FORTRAN compatible list processor language. I did that while I was at G.E. and I did the work on the CDC 1604 that belonged to CDC down the street. We actually rented time on the CDC computer. Crazy thing. Here I was working for General Electric. They let me do what I wanted and I put in a budget which included machine time. Terribly sloppy management. Anyway, SLIP became quite famous and a very good thing. I am sure it is still used in many places.

RM: Was it actually packaged by G.E. and sold?

JW: No, no. I cooked up the system which was, so to speak, a direct competitor to IBL-5. IBL-5 had certain troubles; it was a free standing system. If you were at a computing center and wanted to do something in IBL-5, you had to dismount the FORTRAN tapes, mount the IBL-5 tapes, and the new monitoring system and everything was different. This worked fine in places like RAND that were committed to IBL-5 to some extent, but it really couldn't work in a university computer center, for example. Similarly with LIST, which McCarthy did at about the same time.

What I did is to cook up a set of FORTRAN subroutines that made list processing in FORTRAN and in FORTRAN notation possible. This meant, among other things, that people who wanted to do list processing could do it within the FORTRAN monitor system without ever telling management that they were doing anything unusual. It didn't require remounting tapes or anything of the kind. It became a fairly powerful teaching vehicle for teaching people about list processors. It is a hell of a lot easier to go by way of FORTRAN,

which they already understood, add a few things and say that is how you do it. So it became very popular.

It was implemented in many, many places and, as I say, it still exists today in places. I published a paper on it in the Communications of the ACM in 1962 or 1963 called Symmetric List Processing. I did a very careful job of describing it in sufficient detail so that anyone could implement it just from the paper. So other people reimplemented the thing and I gave it away. I think I might have started a business on that basis to maintain it, and to run it, and to sell it, and all that sort of thing, but that never occurred to me so I didn't do it. I just published it. Which, among other things, helped to establish my fame in certain circles of the computer community. It became a quite famous system. That or my general fame, got me invited to MIT, I guess.

RM: Let us just get a date on SLIP.

JW: Yes. I published either late 1962 or 1963 in the Communications of the ACM.

RM: So you were working on it in 1962?

JW: Yes. As far as my personal history is concerned, in the summer of 1963 I was invited to participate in a Project Mack Planning Conference. After that summer I was asked whether I would join MIT for a year as a visiting professor, which I did. After that year, I was asked to stay. That is where I have been ever since.

RM: Okay, as far as my key area of interest we are getting late, but I would like to go on and talk a little bit about your work at MIT, and talk about ELIZA.

JW: I became involved with a time sharing system at MIT, the CTSS, and in retrospect, I can't think of any important or indeed unimportant contribution that I made to it. It was fun and I think it is probably fair to say that what I did mainly was to take advantage of it, and in a certain sense become sort of a pioneer user of it, which is to say really a pioneer user of interactive computing. One of the first things that happened when I got there was that I got a console at home, so I worked at home a lot.

RM: Not in the bedroom?

JW: Not in the bedroom! No. It was widely known that it was in the bedroom, but it wasn't. I was able to feed back to people actually working on the insides of the system, that was one of my experiences in using it. I think for a while I became an ambassador of the system. I would go around giving talks on how good it was, my experiences with it, and generally implying that there really isn't any computing other than interactive computing, and all that. In a way, I suppose, I earned my keep with respect to that system.

There are a number of ways to tell how ELIZA came about. They are all different and they are all true. That's life. Life is not single dimensional and everything has quite a number of different reasons, which may even contradict one another, but they are all true anyway. One way to describe how I got into ELIZA is the following.

I got this console installed at home and since I have kids, one of the first things I did was to rig up a little question-answering system. It had the property that I could type in any

question that could be answered yes or no and generally speaking it would always give the right answer. That was, of course, magic. After a while people would form all sorts of hypotheses about how this was done; depending upon the number of characters in the sentence, or whether it began with a W, or God knows what; all sorts of conjectures. Whatever conjectures people made I could always demonstrate that, no, that isn't it.

The trick was that if I wanted the answer to be yes, I just typed in whatever the question was and the computer paid no attention to that at all. If I wanted the answer to be no, then I would hit backspace on the typewriter first before I typed anything else. Since the carriage was already all the way back, you couldn't tell what I was doing, but the character was in fact recognized by the computer, which would then patiently listen to the question, so to speak, and then type "no." [Laughter] This baffled people because there was absolutely no way of telling what I was doing. You couldn't look at the paper because the backspace didn't show. They would finally say, "All right, let me sit

down and do it." By being moderately clever I could con them into asking only questions that could be answered "yes." [Laughter] It completely mystified them, which is the sort of thing that stage magicians and fortune tellers do. That's all it is.

That got me thinking about how this little program which was all of four or five lines long could be improved. You know how it is with programmers, they are constantly improving what they have already done. I made it a little fancier and still fancier, and finally came up with the idea of fundamentally trying to imitate the way we understand sentences in a context. I developed a system for manipulating words and sentences and all that sort of thing, which I could then use to implement this idea of understanding things in context. I then had a manipulation system and the question was what is the context to be.

I chose the psychiatrist, as I explained in the paper, because the psychiatrist is presumed to understand even when he doesn't. We attribute understanding to him because if it is clear that he appears not to understand, we assume

that he has some reason for making us believe that. If you say to him, "I took a square root of a number," he may not know what you mean, but he might say, "That is very interesting," and you think that the reason he isn't questioning you about it is because he is trying to draw you out or something like that. It is fundamentally for that reason that I chose the psychiatrist thing, Doctor as I finally called the program. That was a good guess in that it really worked. People would attribute understanding to the computer when it didn't.

Now there is an interesting by-play that comes in here (that I will probably remove from the final transcript, but I'll say it anyway.) I was at MIT now, but I had been here while working on SLIP and list processing and all that sort of thing. I had met Ken Colby. Do you know Ken?

RM: No, I don't.

JW: He's probably a guy you should talk to. Well, I'm not sure. You can decide for yourself whether you want to or not. Kenneth Mark Colby is a psychoanalyst who was once a fellow here and got

fascinated by computers. When I started working on this DOCTOR business, I communicated with him and told him what I was doing. Occasionally I would come out here for one reason or another and he was fascinated by this business. I explained it to him and he even got some of his assistants and students together, gave them a little lecture and showed them in effect how to implement it. Ultimately I wrote a paper about it [ELIZA -- A Computer Program for the Study of Natural Language Communications Between Man and Machine," J. Weizenbaum, Comm. of the ACM 9, 1 Jan. 1966, 36-45.] which again, as is my style, made it possible for anyone to implement it just by following the recipes given in the paper.

RM: That is this paper here?

JW: No, it is a paper that preceded that one. This one, in an important sense, summerizes the other one, so you don't need the other one unless you wanted to duplicate the system. ["Contextual Understanding by Computers," Joseph Weizenbaum, Comm. of the ACM, Vol. 10, #B, August 1967, 474-480.] That resulted in a kind of repetition of an earlier experience I mentioned to you with

Burroughs. Ken cooked up a system of his own, essentially an exact copy to mine, which is perfectly all right because I think that is the way the business should work. The trouble is he forgot to tell people that it wasn't his, and to this day, as I discovered when I came here and one of my children took a psychology course, in the psychology book there is a reference to that program and it just mentions Colby, it doesn't mention me at all. Anyway, that started a controversy which was fairly well contained, but there are enough people in the computer industry who know about it that it might surface in some of the discussions you have with people. That is essentially the reason why I mention it.

Ken submitted some papers to some fairly important conferences on this work, which were rejected because the referees happened to know of the work I had done at MIT. It isn't a question of independent invention in two different places, it is precisely the same work. In fact, the way the machine responded, the very same words that my machine used, and so on, were used in his work.

In a way ELIZA determined my subsequent history to this day, and in particular in the following way. There were three reactions to ELIZA that set me to thinking, thoughts that I haven't finished yet. In effect that set doing what I am trying to do here and in some sense motivated the other paper, the Science paper on the impact of computers on society.

The first one was the reaction that particular work had in the psychiatric community. We had the system running at MIT and it got to be fairly famous fairly quickly. People would come and sit down and talk to it. You know about the system, don't you?

RM: Yes, I've played with it.

JW: Okay. Among others, many psychiatrists came in and looked at it and a rather common reaction among many of them, although by no means all of them, was that this was the dawn of automatic psychiatry. Ken Colby, in particular, in a paper that he published on his version of the work, spoke about the possibility of replacing psychiatrists in state hospitals with this system, treating many hundreds of people simultaneously

and all that. That shocked me. If for no other reason than I wondered what the psychiatrist's image must be of what he himself is doing, and whether he thinks of himself as fundamentally responding mechanically to his patients, and so on and so forth.

RM: That is disturbing.

JW: Secondly, what his image must be for his patients to permit him to entertain that kind of image of therapy. That set me to wondering about the whole business of a mechanical conception of man and all that sort of thing.

I noticed almost invariably how quickly people became emotionally involved with it, and how deeply they became involved with it. The story I tell in that paper, for example, is about my secretary, who asked to be permitted to play with the program. She had seen me work on this thing for a couple of years and she must have known precisely what a mechanical device it was. So I sat her down at the typewriter and she started. After about two or three interchanges, she turned around and asked me to leave the room. This was by no means atypical. Absolutely not atypical.

Another event is that the thing existed on a time sharing system, so lots of people played with it unbeknownst to me. You could access it from anywhere, and often people would come to me with transcripts. "Look, here is what I said and what the machine said." I would learn from these transcripts, I would learn weaknesses and be able to fix them up so the thing got better and better. Well, it occurred to me that perhaps I ought not to have to depend on people bringing me these things and that I could fix up the system so that whenever anyone accessed it, the record of their conversation would be left in some file of mine. Then, in the morning, I could just get all those things printed out and look at them. I discussed this with a number of people and it just caused a sensation; I was invading the innermost intimate thoughts and all that sort of thing. The idea that people entrust the computer, or are even motivated to discuss intimate thoughts with a computer, again just sort of shocked me.

Finally, a number of people, particularly educators, came to me and said they needed help

particularly with computer aided instruction systems, and they came to me particularly because I had "Solved the natural language problem with respect to computers." What I read into that, I think accurately, was the extent to which people attribute things to computers, particularly in general to technology and they just don't really understand. They are almost bound to overshoot and attribute much more power and so on to these things than they actually have. I started wondering about the general effect of this. The general public hears about computers navigating airplanes, computers doing psychiatry, computers doing this, computers doing that, how can they make decisions about computers when their view of them is so terribly distorted and generally speaking, exaggerated? Of course, the press has a lot to do with it, as well as the propaganda emanating from the computer industry itself. That's a sentence from the Science paper. That has a lot to do with it. In particular, for example, I am thinking of the article that appeared in Life Magazine in 1970 about Shakey the Robot. Do you recall it?

RM: Yes.

JW: Well, what are ordinary people to do? They don't have any way to accurately measure what computers, and more generally, what technology is all about, what it does and what its limits are. I started worrying about that, and worrying about it in a quite serious way. My first attempt to put anything down on paper about that is that Science paper. What I am doing here is essentially digging very much deeper and trying to establish a position fundamentally having to do with the proper place of computers in society. I started out trying to determine some limit to artificial intelligence, and I decided that was essentially a sterile effort. If one is going to talk about limits at all, then one has to say limits of principle which means that one might be talking about a thousand years from now, two hundred years from now or whatever, and that is really not what bothers me. It really doesn't. I'm not really bothered about what might happen three or four hundred years from now, I worry about what is happening now and what is likely to happen in the next ten to twenty years. So I really don't want to concern myself too much with ultimate limits

and all that sort of thing, but it is a topic that is hard to stay away from. What I'm thinking about is much more questions having to do with what it means to trust the computer. What would it mean, for example, if one were to entrust judicial decision making to a computer. Questions of that kind. I am in the middle of that work and so I would rather not talk very much about it.

RM: It is an incredibly fascinating subject, and I could just talk all day.

JW: Yes, and I have gotten terribly, deeply involved.

If you just look at the kinds of books that I have here that I have had to look at: Social Dimensions of Law and Justice, Phenomenology of the Social World, Freedom and Control of Man In Society, The American Mind, God and Golem Incorporated by Weiner, The Psychoanalytic Study of the Child, all this stuff is extremely relevant. It just does on and on. It turns out that it is an incredibly deep well that I have fallen into and I have no idea when I can climb out.

RM: All that from playing games in your house with a computer.

JW: Yes, it all started out with that.

RM: That is very intriguing.

JW: At least that is one way of telling it. I think that there must be a large number of other ways of accounting for the same thing. For example, I mentioned that I picked the psychiatrist as a way of doing business with ELIZA because one attributes understanding to the psychiatrist even if he doesn't understand. Well, would anyone have picked the psychiatrist? Or wouldn't it take someone who already knows a little bit about it and in some sense sympathetic to that form of human relationship and so on? So there are clearly other reasons that come out of my background. Certainly not everyone was shocked to notice the things that I noticed. About the reactions to ELIZA. Clearly there are other explanations that are equally valid, but those are the ones I'm stuck with.

RM: Yes. I must say that they give me some pretty wierd feelings in my stomach, too.

I think what we have to do is cut now because I must go and see Paul Armer.

Thank you very much indeed.

END OF INTERVIEW

Index - Joseph Weizenbaum
17 April 1973

Advanced Research Projects Agency
(ARPA)

33
Armer, Paul
80
Association for Computing Machinery (ACM)
66
Bank of America
35, 38, 39
Bendix
26, 28, 32, 34
Bendix Digital Differential Analyzer
9
Bendix G-15
28, 29, 31, 45
Bendix G-20
32, 33, 34
Brown, Gordon S.
9, 11
Burroughs
2, 3, 35, 73
Bush Differential Analyzer
1, 2, 17
CDC 1604
64
Cinema Integrgraph
1, 9, 11
Colby, Kenneth
71, 72, 73, 74
Compatible Time Sharing System
(CTSS)
67
Computer Control Company
9, 11, 24, 25, 26
Computer Research Corporation
48, 64
Dartmouth College
12
Datamatic
25
Datatron
39
Doctor
71, 72
ELIZA
67, 68-77, 80

ERMA
35, 36, 37, 40, 55, 56, 60
Evans, David C.
26, 27, 32, 33
FORTRAN
64, 65
Fein, Louis
11, 12, 22, 24
Frankel, Stanley
28, 40
GE 225
52, 53
GE 235
62
Geiser, Kenneth
59
General Electric Company
7, 25, 33-38, 40, 42, 44, 49,
50-52, 55-58, 60-62, 64, 65
Green, Sid
35
Grosch, Herbert R.
42
Hanson, John
25
Herold, Henry
36, 40, 55
Hewlett-Packard
28
Honeywell, Inc.
24, 25
Humble Oil Company
31
Huskey, Harry D.
2, 3, 26, 27, 28, 30, 31
IBL-5
65
IBM 650
3, 22
IBM 704
42
IT&T
21
Information International
Incorporated
25
Intercom
30, 31
Intercom II and III
32
International Business Machines

34, 39, 50, 51, 56, 57, 60, 61
Jacobson, Arvid
1, 2
Johnson, Robert R.
35, 37, 38, 41, 44
Katz, Charles
59
LGP-30
28
LIST
65
Lasher, Claire
58
Leventhal, Jay
36, 40, 42
Massachusetts Institute of
Technology
1, 9, 11, 30, 63, 67, 71, 73,
74
McCarthy, Joseph
65
McCracken, Daniel
42, 53
Mean Free Up Time (MFUT)
12
Microprogram Control Units (MPCU)
47
Monroe
6
NCR 304
44-49, 56, 61
National Cash Register
44, 45, 46, 47, 58
New York Times
57
Nolfry, Lou
59
Packard-Bell
32
Palevsky, Max
9, 32, 36
Pavenon, John
36, 40
RAND Corporation
65
RAYDAC
12, 17, 19, 22, 24
Radio Corporation of America (RCA)
50, 52
Raytheon
12, 24, 25

SLIP
64, 66, 71
Shakey the Robot
77
Spielburg, Arnold
52
Stanford Research Institute (SRI)
26, 35, 37, 39, 40, 41, 42
Stanford. Center for Advanced
Study in Behavioral Services
1
Strickland, Harold A., Jr.
62
UDEC
2 thru 7
UNIVAC
59
US Army
43
US Naval Air Missile Test Center,
Point Mugu (USNAMTC)
11, 12, 19, 21, 22
US Navy
12, 19, 22, 24, 49
Vietnam
21
Wall Street Journal
57
Waller, B.F.
25, 26
Watergate
21
Wayne Universtiy
1, 2, 9, 11
Wiener, Norbert
9, 79
Zeaheb, Dave
41
Zipf, Alfred R.
57