# Computer Oral History Collection, 1969-1973, 1977

**Interviewee:** Paul Armer
**Interviewer:** Robina Mapstone
**Date:** April 17, 1973
**Repository:** Archives Center, National Museum of American History

MAPSTONE:

The date is April 17, 1973 and this is Bobbi Mapstone. I'm talking with Paul Armer at the Center for Advanced Study in the Behavioral Sciences in Stanford, California. Let's start by going back to the beginning.

ARMER:

Okay. In high school I had represented my school in exams conducted by the American Chemical Society. That year the exams were being conducted by the head of the Chemistry Department at Loyola University in Los Angeles, where I was headed as a college student because they'd decided to give me a scholarship. Allard, who was head of the department, came up to me at the end of the exam and said, "You are going to be a chemistry major, aren't you?" For lack of anything better to say, I said, "Well, I suppose so." Indeed, when I started at Loyola, I registered as a chemistry major.

I soon went into the Service, got educated in meteorology at Uncle Sam's expense, got back out of the Service, re-enrolled at Loyola as a chemistry major. Only at this point in time, I had more than enough credits for a degree, but was still taking freshman chemistry. [laughter] About this time, my wife got pregnant, and since I assumed that I was six years away from a Ph.D. in chemistry, it seemed like it would be a good idea to look around for some other ways of earning a living.

I went to work for United Airlines as a dispatch clerk, hoping to become a dispatcher. The reason for this being that while I'd been in the Service. I had worked as a flight operations officer doing the job that a dispatcher does in a civilian airline. In order to be a dispatcher for an airline, you have to be licensed by the Federal Aviation Agency, as I guess it was called in those days. I went off and got my so-called ticket, came back and informed my boss that I was now ready for an assignment as a dispatcher. I was informed that nobody in the organization moves any faster than the people ahead of him on the seniority list of the company, and it would be a long time. So I said, "Thanks," and went looking for another job.

One of the places that I went looking was the Bureau of Occupations at UCLA, where I had gotten my degree. They sent me off to a place called Project RAND, which at that point in time was a part of Douglas Aircraft. They had openings for mathematicians and

it sounded sort of interesting, although they claimed they couldn't tell me an awful lot about the job because it was classified. The people seemed interesting and they paid a bit more money than I was making at United Airlines, so I said, "Okay, yes," and hired on as a mathematician. It turned out that meant grinding a desk calculator. That was in July of 1947.

That fall I learned that RAND had ordered some accounting equipment from IBM in order to do scientific computing. The guy who was pushing the use of IBM equipment for this purpose was Cecil Hastings. I went to Cecil, who had hired me in the first place, and said, "Hey, I would like to learn something about the use of this equipment." He said, "Fine." I started working on swing shift when the equipment came in.

At this point in time, Cecil was doing, what we called in those days, procedure writing, which was what we would call programming today. He wrote out lengthy written instructions as to what happened at each step in the processing of a deck of cards. Cecil usually arrived at the office sometime between 10 and 12 in the morning, and would go home sometime between 2 and 4 the following morning. He was a very unusual guy. The nature of the work was such that there would be times when the machine was grinding out one response every thirty seconds or so, so if you had a deck of several thousand cards it would take a long time to get that deck through the machine, so we sometimes had time on our hands. One night I went to Cecil and said, "You need some help. You're working much too long hours. I have time to burn, and further, I'm convinced that I could do some of these things very well. They're not particularly difficult." He said, "Okay." So I started for a while both running the machines, and, when I had time, helping Cecil by writing procedures.

After a while, they also asked me to head up the central pool of desk calculator operators at RAND and to simultaneously move to the day shift to do that and to help Cecil on days, since that was the time when we could interphase with the customers for whom we were doing this computing. In some sense, Cecil was the first programmer at RAND, and I was the second.

After awhile, Cecil moved off into his own research and I became head of the programming staff, which had grown somewhat. I managed to get rid of the responsibilities of the desk calculator operators by convincing management that the function should be decentralized, which I think was the right thing to do. It had gotten large, so in an economic sense it was much wiser to disperse it over the various departments.

There were lots of other things going on at RAND. Let me, for the moment, stick to what was happening to me in the organization.

In 1952, in the words of John Williams, "George Brown had an unfortunate encounter with the stock option," and left RAND to go to International Telemeter. It was eventually Telemeter Magnetics, but it was originally called International Telemeter. Among other

things, they were going to be in the pay TV business, and, in fact, the bulk of their financial backing came from Paramount Studios. At that time I became acting head of the department.

MAPSTONE:

Are we still talking about the programming department?

ARMER:

No, the whole numerical analysis department, which George Brown headed until he left in 1952. I found out about that promotion in sort of a peculiar way. I was at a meeting and some people worked for me as programmers came in and said, "Hey, congratulations!" I said, "Congratulations on what?"

"You're acting head of the department." I said, "Well, thanks for telling me, because nobody else has."

MAPSTONE:

Really? [laughter] The last to know.

ARMER:

From 1952 until 1962, although I forget the exact date. I was head of the department. In 1962, and again I'm not sure of that date, although I guess there's a document and I could check on it, I had become quite concerned that I was becoming obsolete with respect to the field, and that I had devoted so much time to administration that I wasn't really as familiar with the actual technology as I ought to be. I persuaded Colebaum, the President of RAND, to let me switch hats with Willis Ware; let Willis take over the department, and I would become associate department head, which had been Willis' previous title, and I would spend that year in a semi-sabbatic way getting myself refurbished.

I made one basic mistake, I only moved into another RAND building, whereas to do the job properly I should have gotten out of town. The problem being that with a department of 135 people, I forget what the number was, there were lots of people problems. Of the various skills that each of us had, I was probably a little bit better than Willis on the people skills although he was obviously better in some of the other things. Anyway, people kept coming to me with their problems; I couldn't boot them out.

About that time, again, quite by accident, I got quite interested in the social implications of computing. That accident being that the World Affairs Council was having a session in San Francisco on automation and employment. As they did quite frequently, they called up RAND and said, "You must have somebody that knows about this, can you send somebody up to participate?" Whoever took this call at RAND said, "Paul Armer's not

doing anything at the moment. He'd probably be willing to go." So I said, "Yes, I'll go. Obviously I know something about computers and I probably know something about automation." I also felt that this was an issue and indeed one in which I might contribute something. About three months later it suddenly hit me between the eyes that although I might know something about computers and automation, I knew practically nothing about economics; and it was really economics that mattered and not the computers.

I started going back to UCLA, taking courses in economics. And I think that it's been the case that there's been lots of scientists and technologists that got concerned about social implications and extrapolating that indeed they know something about their own science or technology and figure that they really know as much about the rest of the world and go off and say some rather asinine things. Deck Belman, certainly one of the better mathematicians in the world, at one time made the statement that within a few years two percent of the labor force would be able to produce all the goods and services that the rest of the country could consume, and that we were therefore going to have a glut of goods and services in America. [pause] In fact, at about this period of time there was something called the Ad Hoc Committee for the Triple Revolution. The three pieces of the Triple Revolution being what was going on with respect to race relations, what was going on with respect to nuclear energy, and thirdly what was going on with respect to computers and automation. They wrote a letter to the President, and I think in some sense, particularly with respect to the racial issue, they were right. They had picked up this glut of goods and services idea, indicating we're going to have a real problem with leisure within five years. We were just quite wrong on that topic.

MAPSTONE:

This awakening then was around the early Sixties, for you?

ARMER:

Yes, but again, quite accidental. At least the fact that because of the fact that being invited to this conference I suddenly became aware of the employment issue. I think another interesting thing, that wasn't accidental at that time, was that I was concerned about being obsolete clear back in the early Sixties, and that's the topic that I'm attempting to write a book on now.

MAPSTONE:

Oh, really?

ARMER:

In fact, I resigned as Director of the Computation Center at Stanford because I really didn't think I was doing a very good job. My boss, Bill Miller, who was a computer type himself, told me I was out of my head. I said--I'm about to inflict some mathematical

phraseology on you-- "You only see the function, but I'm seeing the first and second derivatives, and I think that five years from now, if I stuck around for those five years, that you'd be unhappy. You may be quite happy now, but I don't like what I see happening to the derivatives." So I went off to Harvard to start on this book.

MAPSTONE:

Oh, that's when you started?

ARMER:

Yes.

MAPSTONE:

Right.

ARMER:

Okay, you're interested in the very early history and not what I'm up to at the moment.

MAPSTONE:

Well, that's interesting too. Let's backtrack and take a look at RAND in the late 1940s and 1950s.

ARMER:

Okay. As I say, in the fall of 1947, we had acquired IBM equipment that had been designed for accounting purposes, and were using it for scientific computing. To begin with, the heart of the installation was a couple of 602 calculating punches, I guess they were called. Later on we got 604s, which used tubes for the first time. The 602s were electrical but not electronic, whereas the 604 was, I guess, the first electronic calculating punch that IBM came out with. Also, about this time, they had come out with a modified 602, called the 602a, and we got some of those in. We decided that they really weren't much of an advancement, particularly since the 604s were much faster and more versatile. We quickly sent the 602As back.

The same kind of a thing was going on at most of the other aircraft companies, and before we got our own gear, we were using the Douglas equipment. At that time RAND was still part of Douglas; we were Douglas employees working on Project RAND. People who were assigned to the Project RAND payroll used the Douglas gear on swing shift. Don Madden was one of those people.

When we got our own gear they moved over to the RAND premises, and they pretty much became the people who worked the day shift. We manned a swing shift of one supervisor, who indeed knew something about such gear, plus myself, and some other people who were in some sense trained.

About this period of time, some of the other aircraft companies were getting involved. I think Convair had been doing something with IBM equipment probably before anybody else in that part of the world.

MAPSTONE:

Oh, really? Do you know what--

ARMER:

Ben Ferber is the name.

MAPSTONE:

I've heard the name. Is he still around?

ARMER:

Gosh, I haven't heard of Ben in a long time.

It was about this time that Northrop was lashing up a tabulator and I guess they modified a 603. I don't think they called it the Card Program MED Calculator, I guess it was IBM that invented the name, but Northrop sold the idea to IBM.

MAPSTONE:

They called it Betsy.

ARMER:

Yes. Northrop sold the idea to IBM, and IBM decided they would build it. [Interview interrupted by visitor.]

MAPSTONE:

We were just getting into the fact that Northrop had set up the CPC.

ARMER:

By that time we had an IBM sales representative. His name was Chuck--

MAPSTONE:

O'Malley?

ARMER:

No, O'Malley was around. He was Chuck's bass. He was killed in an airplane accident about ten years ago. We heard from him that IBM had decided to market the Card Program MED Calculator. We got several of those in; I guess we eventually had three. That made a real difference to computing around RAND.

Before the CPC, every job that we did for each case that you were calculating you did one step on all of the cases, so you didn't know the answer to one case until you knew the answers to all of them. This meant that you had to decide in advance before you knew the answers to any of the cases, what all the interesting cases were. It was really a very inefficient way of doing things, because you were almost always surprised.

With the Card Program MED Calculator, although it wasn't really a stored program machine, you laid out your program, and you would take one case from beginning to end before you started the next one. As a result, you did many fewer cases. We were having a hell of a time getting enough people to do the programming for the jobs that we had around, so we came up with this bright-eyed notion of a do-it-yourself system for the CPC, which we called Combo-Math. The CPC was often called The Combination, because it was a combination tabulator calculator. We put out this manual wherein we said you can do your own computing now, and you don't have to wait until we have time to assign a programmer. All this did was inundate us with requests for work, because we really hadn't thought it through. Sending out this memo and distributing copies of this manual as we did, was in a sense advertising that we had a new capability.

We were also rather naive about how easy it would be for other people to use it. In other words, it looked easy to us, but it wasn't that easy. Some people just said, "Hey, I've got this job I'd like you to do on the CPC." The demand for programmers grew, grew, and grew at RAND.

Simultaneously with these developments, it hit us that we would never do the amount of computing that we thought we were going to want to do with machines with the capability of the CPC. This led to John Williams, George Brown and Bill Cunning going around the country doing a state-of-the art survey and attempting to answer the question, "How are we going to meet RAND's computational needs in the future?"

MAPSTONE:

Before we go onto the trip, what were the things that were putting the demand on RAND? What kinds of programs were there in house?

ARMER:

Probably the biggest job that I can recall in the very early days was something called the Bomber Study; an attempt to optimize the design of airplanes. This resulted in somewhat of a disagreement between RAND and the Air Force, because the study showed that they could get many more airplanes if they'd settle for turboprops. However, the Air Force wanted jets, and I suspect in retrospect that they were quite right.

One of the things which made working at RAND exciting, particularly in the computer end, was that they were into all kinds of interesting things. Each of the various departments came to RAND with their problems. I can remember early on working on a fairly large study of the incidence of mental health in the U.S., which resulted in a book by Marshall and Goldhammer.

MAPSTONE:

What was the title of the book? Do you recall?

ARMER:

No. If you're really interested, I can find out quite quickly. [pause] Everyone at that time believed that mental health was getting to be much more of a serious problem than it had been. Their study showed that things were really quite stable, but we were treating more of the people than we had in the past. Like other things, when you begin to pay attention to them, you then uncover more of the problem, and if you just look at the numbers being treated, you can conclude that the problem is getting worse. It's called The Frequency of Mental Disease: Long-term Trends and Present Status.

MAPSTONE:

Okay.

ARMER:

RAND-R 157. Out of print as a RAND-R.

MAPSTONE:

Is that a listing of RAND publications?

ARMER:

Yes. Volume 1, 1946 to 1962. I doubt that the original Williams-Brown report is here, undoubtedly it was before RAND was assigning numbers to such things. It was probably an internal memorandum.

Williams, Brown and Gunning went off to see what was doing.

MAPSTONE:

Gunning, of course, was a hardware type.

ARMER:

Yes. To back up a little bit, RAND was also active in the analog computer arena, and I think made some important contributions there. Early on we got the Reeves analog computer called the REAC, which we proceeded to rebuilt to the point that really all that was left-over from the days that it came in from the manufacturer was the nameplate.

The original gear was like a telephone switchboard and when you wanted to plug-up a problem, you came with long telephone patch cords and you plugged in your problem. Then, if you wanted to go off and adjust the results and turn the machine over to somebody else, you had to pull out all your wires and then he plugged-in his own. Then if you decided that there were some other interesting cases, you had to plug it all back up again.

In the next room we had machines with IBM plug boards, and we said, "We want to marry that technology to this machine." At that point in time, this was before the Consent Decree of 1956, IBM refused not only to sell the machines but also their parts. We wanted to buy some of their parts and I think we were the first organization to break through that.

We got IBM to agree to sell us some of their parts; they sold us a plug board. To get that done required intervention from the Air Force Chief of Staff, I believe, to Mr. Watson.

MAPSTONE:

Pretty high power, isn't it?

ARMER:

In some sense, that was getting the camel's nose under the tent, and I think brought to the attention of other people in Washington the fact that there were problems with IBM's policy.

If you wanted to input an arbitrary function into the REAC computer, you would plot the function that you wanted on a piece of graph paper and mount it on the drum. Then

someone would have to sit and turn a crank which would keep a pointer on this curve as the machine rotated the drum. We decided that was for the birds, because not only was it expensive in the sense that it took somebody sitting at each one of these tables, but it was just damned inaccurate. So we came up with a scheme for reading these, which was then adopted by the industry.

I'm trying to think if there were any other significant developments in respect to the analog field. It was because of the analog computer that Gunning was around.
At this point in time, too, the Institute for Advanced Study at Princeton had the construction of their machine under way. The choices in terms of how were we going to meet the problem of fulfilling our computational needs were: are we going to build another machine like the Princeton machine, or are we going to buy a copy of the machine that Eckert and Mauchly were building for the census group? The conclusion was that the Eckert and Mauchly machine, since it was serial, was not going to be fast enough for what we wanted. The decision was to build a copy of the Princeton machine.

MAPSTONE:

This of course was after IBM had agreed…

ARMER:

IBM had said that they had no plans to build a machine. We like to think that because we had shown this much interest in getting a machine that we had some impact on IBM. In fact, I think you told me George Brown told you the story of his watch.

MAPSTONE:

Yes. I did. One can only conjecture that since Watson did invite George Brown meant that RAND had some immeasurable effect on the settlement of the 701 decision. But I do think it's immeasurable.

ARMER:

I've got to get into those boxes. I'm sure I have John Williams' curtain lecture on the JOHNNIAC and I think you'll enjoy it. He talks about the IBM unleashing hoards of engineers to come after the digital computer market. [pause]

We decided to embark on the construction of JOHNNIAC. In early memos, I'm pretty sure Willis talked about building it for $150,000. By the time it computed its first prime number, I think we conservatively had a million dollars invested. That was why when Cuthbert Hurd came around to tell us about the IBM Defense Calculator, as the 701 was originally called, and told us that he was going to lease it to us for $8,000 a month, we said, "You'll never do it, but if you're crazy enough to do it, we'll take one!" [laughter]

When IBM came around some time later to say it's going to be $15,000 a month, we said, "We told you so, and we don't want it at that price," and we cancelled our order. We were much too optimistic, not only about how much JOHNNIAC was going to cost, but how soon it was going to become operational, so that eventually we went back to IBM and said, "We'd still like a 701."

MAPSTONE:

Was it $21,000 by that time, or did it stabilize at $15,000?

ARMER:

I forget those numbers.

MAPSTONE:

Yes, I do too.

ARMER:

I think it was probably more like at least $18,000 and maybe as high as $20,000.

Eventually JOHNNIAC came on the air. One of those things which is certainly not known to me, but maybe is knowable to some of you who dig into this, is whether we were the first to do this. Early on we said, with respect to input to the machine, "We want to couple the technology in the IBM gear with our device," we decided that we were going to have punched-card input and output. As I say, whether or not we were the first ones to have that idea, whether or not we transmitted that to others before they came up with it independently, I haven't the slightest idea. But in the mode of the ant riding the log downstream, we thought that was our idea.

Another difference we decided to incorporate in JOHNNIAC was that we would pay a lot more attention to reliability. In those early days, one of the ways we were getting our computing done was to go off to Aberdeen to use ENIAC, and I don't know whether by that point in time we were trying to use SWAC or not, but it just struck us, particularly those of us who were users compared to builders, that the reliability situation was just intolerable. In fact, with respect to SWAC, this might have been after the decision reliability-wise had been made about JOHNNIAC, but in the early days the mean free time between errors was like a tenth of a second! We managed to live with that, but about that frequently the machine would stop and dump everything. Consequently, you built in a lot of routine so that if you had trouble, you could back up and start over and not lose everything. Obviously it would be insane to run a problem which needed 5 minutes of continuous running time to get an answer, because you'd never get an answer!

MAPSTONE:

Considering the speed of computers at that time, which was relatively slow, and the failure time, the down time, it's amazing how you ever did anything. It really is.

ARMER:

I remember making a statement that was obviously very wrong. When IBM embarked on building the machines for the SAGE system, I think the tube count was to be 40,000, 100,000, whatever it was. I did a mental calculation of the mean free time between the errors, extrapolating what I knew about computers in those days, and I said, "It will never run!" The reason that it did run was that IBM literally spent millions of dollars developing reliable tubes. We were really using technology that was around for the radio and radar fields, and there it didn't matter if every once in a while there was a little bit of noise, so you had some static. But in digital systems where that happened, things got screwed up.

The SAGE computers did work. They incorporated a system in the original design whereby they were going to punch out a punched card when there'd been a failure, because they had ways of sensing failure. Then the machine would go back and do the calculation over again. As part of this "it'll never work" prediction, I had said something about it'll never do anything but punch cards, and indeed they had to disable this particular feature because it didn't do anything but punch cards! [laughter]

MAPSTONE:

I think we'd better turn the tape.

[Tape 1, Side 2]

ARMER:

I'm continuing with some of the other things about JOHNNIAC although I would guess you'd probably get better information from Bill Gunning.

We also decided that since the other people who were building Princeton-class machines were all pursuing the Williams tube state-of-the art with respect to memories, that we would try something different. We elected to go along with the Selectron, as developed by RCA. Jan Rajchman was the prime guy behind that. The Selectron memory worked exceedingly well, which I think is due mainly to Keith Uncapher who was responsible for it under Bill Gunning, although I suppose it's quite possible that Bill could have played more of a role than I thought he did. Bill, in some sense, is one of my household gods, and he's the kind of guy who would give credit to somebody else, even though much of it might have belonged to him. However, the Selectron memory was Keith's baby.

We actually built something called JOHNNIAC Jr. as sort of a test bed for the memory. Shortly after they got JOHNNIAC Jr. working, they decided to see how it would run. My recollection is that it ran something like four days before it made an error, and further, this was with Selectrons which were considered rejects.

The Williams' memories at that time had a peculiar read around ratio problem, so the Selectron memory was just very reliable compared to it. The reason that it never got into any other machines was that about this time the magnetic core technology was on the horizon, and RCK didn't like building the Selectrons anyway because their yield wasn't good. Have you ever seen one?

MAPSTONE:

Yes. I've seen it.

ARMER:

It's like trying to assemble an electronic punch-board in a bottle.

MAPSTONE:

It's an incredible thing isn't it?

ARMER:

We saw the fact that the Selectron had no future, and that the future was with cores. In fact, we immediately set out to come up with the specifications for the 4096 word memory for JOHNNIAC; the specifications that you were saying earlier you'd like to get a hold of.

MAPSTONE:

Yes.

ARMER:

And I think one of the contributions of that specification was the incorporation of this philosophy of reliability. Something that I did mention earlier as a way of getting at this reliability was to have marginal checking built into the machine. You could flip switches which would lower some of the voltage levels and sit there and do this marginal checking, and therefore, in that fashion, hope to find hardware troubles that were about to crop up before they did, and replace the parts that were marginal.

Another design goal with JOHNNIAC was that it would be somewhat better human-engineered. At that time you entered most of your data on the Princeton machine, via

punched paper tape; if you wanted to enter a number by hand, you did this with a bit of clip wire. We decided we wanted a decent console with buttons that you could push. I suspect you've got a fair amount of data on JOHNNIAC.

MAPSTONE:

Yes. On the actual development.

ARMER:

I suspect it's probably not reasonable to talk more about that.

MAPSTONE:

I think what we should talk about is how JOHNNIAC was used, some of its significant contributions through maybe language development, its significant outcomes, and how it influenced other areas.

ARMER:

I suppose the most significant thing in that area was the development of JOSS. To get into that you'd have to go back to the fact that in the early days around RAND was a guy by the name of Al Newell. He played a key role in what was originally called the Systems Training Project. Some of the psychologists around RAND were interested in training, and man-machine interaction. They were looking around for a test bed within the Air Force, where there was a fairly large man-machine interaction to which they might bring to bear some of the things which they knew about training.

They decided an interesting area was air defense installations, so we actually built a model--I don't know what the exact name was, but something like the Air Defense Information Center-- McCord Field, in Tacoma, at RAND. There were some interesting computer aspects to that, since we were going to attempt to simulate in some fashion this Air Defense Direction Center, or whatever it was called.

One of the key things in these centers were radar scopes, so the question was, "How do you simulate a radar scope?" We decided to use the carriages off IBM tabulators, and we would have a number of them around the room. First, we would use the tabulator to plot an X on a piece of paper where there was supposed to be a blip on the radar screen. Then, having prepared all these scripts in advance, every thirty seconds or so, corresponding to a rotation on the radar antenna, although we didn't have as many, all these carriages would advance one frame. The operator would sit and look through a piece of glass which had a circular hole in it, so that it looked a little bit like a radar screen, and on this piece of paper he would see Xs where he was supposed to see blips. I mentioned earlier how we had gone to IBM and said, "Hey, we want to rent some tabulator carriages; we

don't want any tabulators, just the carriages." By then we had the reputation within IBM of being a rather peculiar customer, and eventually we got our carriages and built this lab.

The first experiments were run with college students and it was obvious they were doing a much better job of air defense than was being done on the field. Of course, one hypothesis to explain that difference was that we were using college students rather than the kind of people that were being used by the Air Force. In order to find out whether or not that was the explanation for the big difference, we got in--and when I say we, I mean RAND--people from the field who had been doing this. After awhile of going through these simulated exercises, they were doing a much better job of air defense than was being done on the field, which didn't surprise the psychologists at all, because what they set out to do essentially was to improve the skills of these people by training.

Really what they did, since this was a man-machine interface, with an information processing system in the broadest sense of the word "system" they would increase the information processing load on the system by slowly making the problems more difficult, with more planes to keep track of and more of an information-processing load. As this load increased, the organization learned to separate the wheat from the chaff, or the important from the unimportant, and consequently was able to keep good track of the important things and not over-burden the system with the unimportant.

RAND'S normal mode of operation when doing something like this for the Air Force would be to say, "Well, we've developed something; now go get yourself a contractor and use it." This is what they said at this point of time, but the Air Force came back and said, "Look, for you to transfer all you know about this process to somebody else is going to take time, and we want this very badly. We figure that if you do the job, we can actually have it in the field eighteen months sooner than if we try and get somebody else to do it. So please won't you do this?" RAND did some soul-searching and decided they would, and set up the Systems Development Division to do it. Thus began what's now the Systems Development Corporation.

The technological aspect of this scheme for simulating radar sets was obviously not a very good one, and since we now had more resources, since the project had the attention of the Air Force, we let a contract to IBM to develop a CRT display device for the 701. The use of CRTs as displays was not at all new. In particular, I would guess that the first one had been used by Project Whirlwind at NIT. But I think we had something to do with it becoming a commercial product, because we would beat on IBM and then be paid for the development of the device which was subsequently announced as the Model 740. Then we let some other contracts. We had a contract for looking at the CRT display device, this went onto film and then went into other devices which projected, not directly, but took the information which was thus stored on the film and put it back on CRTs so that the people in the labs actually saw a radar scope. Eventually it was a device which just fed information into the radar device so that other than the fact that the picture was probably cleaner than the operator saw when he was looking at reality, he really wouldn't know whether it was a canned exercise or reality.

In the midst of all this, the developments at MIT which eventually led to SAGE were going on. The problem of doing the programming for SAGE had been vastly underestimated. In the spring of 1954, Don Madden, myself, I guess Kapler [first name?] and maybe some other people, went to the headquarters of the Air Defense Command in Colorado Springs at their request to discuss the programming problems for SAGE. I remember the one thing that appalled us the most was that nobody had thought to set aside a machine for program testing.

At this point, the Air Force said to RAND, "Look, you know a lot about computers because you've been in the game for some time; further, you know a lot about air defense because you've been in the systems programming business for a long time; please won't you do SAGE programming for us?" RAND went back and contemplated its navel somewhat, came back and said, "Okay. However, this is going to make the Systems Development Division of RAND a lot bigger than the rest of RAND, and further it's sort of hard to make the case that it's research." We didn't say it was contract programming because I don't think that phrase had been invented then, but we indicated that it was quite different from what RAND normally does, and therefore we were going to set up a separate organization and think about spinning it off.

Don Madden became head of the programming group which went off to SDD to do the programming. Wes Mehlan went along and eventually became president. I remember at the time saying to Kapler, who was I suppose division head of the Systems Development Division that, "Wes is the best technical guy I've got but I'd never make a supervisor out of him." He eventually became the president, although I think the fact that he didn't stay the president very long was an indication that maybe I was right; at least not a hundred percent wrong.

Al Newell had played a quite key role in all of this, but he decided that he didn't want to continue and came back to the computer sciences department. I don't know whether this is simultaneous or a little bit later but he decided that he wanted to go back to Carnegie to get his Ph.D. under Herb Simon. Herb had also been around RAND as a consultant. I agreed to support Al while he was at Carnegie, and thus began the Newell-Shaw-Simon collaboration.

MAPSTONE:

Shaw from where?

ARMER:

Cliff Shaw from RAND. They did a number of things on JOHNNIAC, including the development of the language called IPL-4, which was information-processing language.

MAPSTONE:

What about IPL-1, 2, 3? What were they?

ARMER:

They were all in some sense prototypes, and the key thing about them was that they involved list processing. Cliff Shaw, I believe, says that he got the notion for list processing out of an Australian paper. I don't really know enough about that. Cliff's another guy who would much prefer to attribute something to somebody else than to take the credit for it himself.

MAPSTONE:

It's terrible when everybody gives the credit to everybody else! You never get to the heart of it.

ARMER:

Certainly these people first brought list processing to the attention of the bulk of the computer industry in this country. They did it, not because they were interested in doing research on programming languages, but because they were interested in developing tools for their own work. I say all this as a lead-in to JOSS, because JOSS came out of that same effort and was again the original talk of the need to develop better tools for our own research. Over time the goals changed from being a tool for somebody of the sophistication of a Newell or a Simon, to building a tool for the people who were somewhat computationally naive.

Later on in the JOSS development, Newell was offered not only a full professorship at Carnegie, but an institutional professorship. He was 33 at the time, and we just couldn't compete with that; going from no academic appointment at all to being an institute professor at 33. He decided to become full-time at Carnegie, but he became a quite active RAND consultant, and may still be. Simon was also a consultant for many years. In fact, they used to come out quite frequently. That was the genesis of the JOSS project.

MAPSTONE:

What exactly was JOSS, and what were its capabilities?

ARMER:

JOSS is still in use at RAND. It's a language. It's a complete system for doing small problems on a computer. I'm not sure that I can lay my hands on a JOSS manual. It was intended to be something that the neophyte with respect to computers could use; not for the man on the street, but for someone who was a scientist in some other field, particularly someone who was facile with algebra and had a logical mind.

I think the best testimony to its success in that regard came some years later. ARPA commissioned Institute Defense Analysis (IDA), to do a study of time-sharing systems. They sent a couple people to come out to RAND, and they called me up in advance to say they were coming and would like to talk to users of JOSS. They arrived one morning about nine o'clock and I had assembled a number of our users. I had, I thought I had adequately told everybody what the meeting was about, so I just immediately started to introduce these two gentlemen from IDA who wanted to talk to them. The guy from IDA got up and said that they had this contract from ARPA to study on-line interactive time-sharing systems. One of our guys said, "Oh, I thought you wanted to talk about JOSS." The key thing was that the IDA man didn't understand the jargon at all, but he was one of our big users and he had managed to get to that point without learning the jargon. He probably didn't know that he was doing any programming. [laughter]

It would be interesting to know the interplay between JOSS and BASIC, again the old ant on the log. I think an early visit by the BASIC people to talk with Shaw had a big impact upon what they did. After awhile, you began to find references to JOSS-like systems in the literature. I think what people meant to convey when they used that term, was a system which was easy to learn, and which was somewhat forgiving at least in terms of contrast say with FORTRAN. Because it was interactive, JOSS could do a lot of things like feedback or give error messages which were a good deal more meaningful than most computer systems. It's been copied in the Soviet Union. Whether or not it had an impact on BASIC, which I think it did, the combination of JOSS and BASIC has had an impact on the development of computer language, in terms of paying more attention to what makes it easy for the unsophisticated user to get useful work out of the computer.

MAPSTONE:

Who were the BASIC people?

ARMER:

John Kennery, who had been a RAND consultant and had been around quite a bit so knew what was going on with JOSS. There was Tom Kurtz maybe, and some other people. Certainly Tom Kurtz later on and maybe from the beginning. This was at Dartmouth.

MAPSTONE:

Oh, okay.

ARMER:

Part of the language development history at that point in time was PACT. I think you've probably seen Fred Gruenberger's paper which discusses the fact that before any 701s

were delivered in Southern California, there was a meeting at the St. Ynez Inn, hosted by Blair Smith of IBM--I think it was his idea and he thinks it was my idea [laughter] or at least it was my idea that IBM should pick up the tab. As a result of that, the Digital Computers Association got started in Southern California.

Out of that environment came the Project for the Advancement of Coding Techniques (PACT). They actually built a compiler that worked for the 701.

MAPSTONE:

Was this the first group effort in a language?

ARMER:

To my knowledge. The group effort worked; we produced something. How much impact this had upon the FORTRAN people would take someone like you really digging in to find out. It was sort of like the development of the Selectron. It worked, but soon thereafter there was this other product named the FORTRAN, which was not only better but had the backing of IBM. So although for a while we mounted a project called PACT-lA to build a modified PACT compiler for the 704, that never saw fruition, and rightly so. I'm sure we kept at it much longer than we should have, but it was sort of pride of authorship and that kind of thing that kept us going.

MAPSTONE:

When you say FORTRAN is better, what would be better about it?

ARMER:

The way of expressing your problem in FORTRAN was a good deal easier. It was more English-like, and also you wrote out statements in a linear fashion. You could write quite a bit on one line, whereas PACT really had a vertical format, so to speak, and took many more lines to accomplish a given amount of work than to do that same amount of work in FORTRAN.

MAPSTONE:

This is probably a silly question. I know the difference between linear and vertical as it applies to programming, but if it is linear how does that affect a machine? Does it take less time, use less memory space, or what?

ARMER:

I'm referring strictly to the only scarce resources here, human time, and not machine time. The industry was beginning to change at this time and we were now starting to have a fair

amount of computer power at our disposal so that we could use it to help with the programming problem rather than using it 100% in doing calculations. I guess I don't know enough about the difference to the machine. Other than the fact that the PACT vertical format would take longer to input into the machine, there probably would not be very much difference in the conversion to machine coding. The key thing was how many lines of code did the individual have to write down, and FORTRAN was just more economical in that sense.

While we were out getting coffee, I was thinking about a key role that RAND played in this cooperative effort which began with Blair Smith getting the DCA people together and ended up with PACT. The culture at RAND was different than it was at the aerospace companies, in the sense that we were a non-profit organization without any axes to grind and cooperation seemed a rather natural thing to us. Whereas to the aerospace people, competition was the normal mode. In fact, PACT almost always met at RAND, because we were concerned that if a lot of Lockheed people were showing up at Douglas or vice versa, some eyebrows might be raised. I believe that within some of the aerospace companies there was discussion later about whether or not it was appropriate to be involved in this because somehow it was letting your competitors know what you were doing. It was not common practice to get together to solve industry problems.

MAPSTONE:

In a way this is where the West Coast and RAND really were different, because for all of this, they did get together. I take it that you feel that RAND probably had a great amount to do with this.

ARMER:

I think that the fact that RAND wasn't in this competitive rat race, so to speak, and that to us cooperation seemed much wiser than each going his separate way, played a part in it. I think it was also the fact that within the aerospace companies it was computer people we were trying to get together. These were people who almost invariably were fairly young and who, in some sense, hadn't been steeped in the competitive atmosphere of their industry, so that it seemed somewhat natural to them. There were some obvious exceptions to that in the leaders of these various groups, but within the air frame, aerospace firms people like Frank Wagner and Jack Strong of North American and John Lowe at Douglas obviously had to believe that cooperation was better than each going his own separate way. However, it was rather unusual for competitors to get together and try and solve problems.

We began by getting together the DCA, and then came PACT, and I think everybody was pleased with the way it turned out. There had not been a cooperative effort with the 701, although IBM had sponsored at least one, if not two, meetings of 701 users at which all the users got together. Some months in advance of the deliveries of the 704, a number of people in the L.A. area were saying. "It's foolish for everybody to go their own way, let's

get together and try to develop some standards and maybe we can do a better job than IBM of turning out software for ourselves, and further, maybe collectively we can have a bit more clout with IBM to get them to do what we want." Thus was born SHARE; and thus was eventually born GUIDE.

MAPSTONE:

How did GUIDE come about? I heard a slightly different story; that actually this was IBM'S request.

ARMER:

Gee, I don't think that's true at all.

MAPSTONE:

I believe that was hearsay.

ARMER:

RAND Was not in the commercial data processing arena to any extent at all, but North American was with a large payroll that they were running. Further, the North American people had been very active in SHARE; Jack Strong being the first president. It was quite natural for the North American people to say, "What we need is a similar organization of users of the 702 and the 705 (GUIDE)." The key actors at North American, as far as I'm aware, were Jack Strong and Ed Law. They began beating the drums for the need of such an organization, and came to me and said, "How about helping us get GUIDE started. In particular, if we can get you onto the program of what eventually became the DPMA, how about going up there and essentially giving a sales pitch for a SHARE-like organization?" I think you have my paper of the eulogy to SHARE?

MAPSTONE:

No, I don't.

ARMER:

No? Well, I'll get you a copy before we leave today.

MAPSTONE:

Okay.

ARMER:

The purpose for which it was written was to help get GUIDE started and this was shortly in advance of the first GUIDE meeting. IBM had certainly cooperated in making a list of 702--705 customers available to the people at North American, but I think that all of the activism and idea came from North American. That may be in that particular document.

MAPSTONE:

There's mention of a "Eulogy to Cooperative Effort." The first meeting was December 1956.

ARMER:

Are you interested in stories which are sort of interesting if you were there at the time, but… I don't know whether they're particularly important with respect to history.

MAPSTONE:

Yes.

ARMER:

I mentioned earlier that we had given money to IBM to develop the CRT display for the 701. One of the conditions under which IBM took the contract was that we would not publicly talk about the fact that we had it until IBM released us to do so. Sometime after we had let this contract. IBM announced that they were going to have a symposium of 701 users and invited people to submit papers. I wrote back and said I would like to give a paper on the CRT device. IBM wrote back saying they hadn't quite decided what they were going to do; whether they were going to have a product themselves or whether to build one for us and forget it. They said, "The decision will undoubtedly be resolved by the time of the meeting, so why don't you go ahead and prepare the paper?" So I did.

I arrived at Endicott the Sunday evening the conference was to begin, and asked G. Truman Hunter, who was the IBMer running the meeting, where I was in the program. If I was on first thing in the morning I wanted to run to my room and do a little more rehearsing; if I was going to be on Wednesday, then I could relax for 48 hours. Truman's answer was, "Well, we haven't quite decided yet what we're going to do about that device, and consequently you're not on the program at all at the moment. If we decide to release the information, then you can talk on Thursday. I guess I swore a little bit and said, "Okay."

Standing alongside of me was Gene Amdahl, now the president of Amdahl Associates, who was the main architect of System 360. At that time he was the architect of a project called the 701A. Gene said to Truman, "Well, where am I on the program? I've got the same question as Paul."

Truman said, "Well, the answer to you is exactly the same as it was to Paul, Corporate Headquarters hasn't made up their mind yet about what they're going to do. If indeed you're on the program, you'll be on for Thursday."

MAPSTONE:

I'm going to have you wait a minute.

ARMER:

Okay.

[Tape 2, Side 1]

MAPSTONE:

You're on, Paul.

ARMER:

Sure enough, the next morning when IBM passed out what the program was to be for the week, I wasn't on the program, nor was Gene Amdahl. But IBM was doing something else at this meeting that they hadn't done before. They decided they would ask everybody for a copy of their paper in advance, and then after the person gave the paper they would pass out the copies. Of course, IBM, in the usual fashion, had to give everybody there a folder in which to put these papers, and they also gave them an index. They had purged the titles of these two papers from the program they handed out, but they hadn't purged them from the index. At this point in time, at least as far as the outside world was concerned, Cuthbert Hurd was Mr. Computer at IBM. However, Cuthbert was ill at the time and Truman Hunter was running things. Truman, instead of saying, "Hahaha, it's a big laugh on us!" attempted to ignore the existence of this index and refused to admit that these two papers existed. At this point, Herb Grosch, who referred to Truman as "Truman the Subhuman" [laughter]--Herb often had unkind things to say about people - drafted a telegraph to Watson which said, "We the undersigned, having very good discussions at Endicott of much value to all of us and we're sure of value to IBM, would like to request that the two papers scheduled for Thursday by Paul Armer and Gene Amdahl be moved earlier in the week so that we can have some time to discuss them and give appropriate feedback to IBM on our reactions." Then he got everybody at the meeting to sign the telegram. Well, this represented really the bulk of IBM's computer customers at the time, so indeed the existence of the 701-A and the CRT device was admitted to, and the papers were given.

MAPSTONE:

The 701-A…

ARMER:

Became the 704.

MAPSTONE:

That was really where it was announced, so to speak.

ARMER:

Yes, somewhat unwillingly by IBM. [laughter]

MAPSTONE:

Bit of blackmail. Pressure, anyway.

ARMER:

Computer customers were something that IBM had a lot of difficulty dealing with for awhile. My characterization of the relationship between an IBM installation manager--I'm talking about the guy who works for, say, Lockheed and manages their tabulating and accounting installation equipment--and the IBM salesman, was that the salesman not only knew more about his gear than the installation manager did, but he knew more about the applications of that gear in the installation. In most instances, that installation manager got the job because a vacancy had appeared and the IBM salesman had said, "Hey, I know Joe Blow who works for ABC Company and if you just call him up and say that you're looking for a supervisor, and he just might come." He made sure that the company somehow didn't realize that IBM was stealing their people away, or helping the guy go somewhere else, but also making sure that the individual knew that IBM had played a key role in his getting promoted to this key job. Consequently this guy always felt an obligation to IBM.

Every few years, IBM would bring him back to Endicott, bring him up to date on what was the latest in their gear and in applications, and always treated him very nicely. This was kind of part and parcel of their scheme for making sure that nobody went someplace else. Further, when IBM announced, say, a new tabulator, the salesman would call on the installation manager--let's call him Joe--and say, "Hey, Joe, I think you ought to replace those two 403 tabulators with a couple of 407s because they'll do a much better job on payroll. In fact, you can get this additional report out." Joe would say, "Yes, that's right," and the salesman would help Joe prepare his pitch to management as to why they should get this new gear. Eventually the gear would be acquired. The customer was, so to speak, always a rather docile customer. Then came the computer people who, at least in those days, almost always knew more about the gear than the IBM salesman.

MAPSTONE:

Right.

ARMER:

This was really very true in the days when people were using IBM equipment designed for accounting purposes to do computing. The epitome of this is what Northrop did; they rebuilt the equipment. That was true of many installations that were using accounting gear for calculating purposes. They knew more about it than IBM did and talked back, they just weren't about to swallow the kind of paternalistic attitudes that IBM had towards their customers. For a long time they just didn't know how to deal with this kind of customer. Cuthbert Hurd's reaction to that, I think, was extremely shrewd. He set up what was called the Applied Science Division, and hired a lot of Ph.Ds, and brought IBM's knowledge of what computing was all about up to the point where it matched that of customers. As time went on and they got many more customers, their knowledge was much greater than that of the average customer.

MAPSTONE:

There was a period there, where if some other companies had some good machines they perhaps could have edged in, because IBM at that point wasn't totally psychologically geared up to handle computer customers.

ARMER:

Neither was Univac Remington-Rand very well geared up. I mentioned earlier that we had used the 701 for producing problems for the Systems Training plan, and we were soon outstripping the amount of time that we had. We needed another machine. IBM had only built 19 701s, and they weren't about to build any more. But we needed another machine now. We couldn't wait for the 704, and even if we could have got the 704, it would have meant a certain amount of re-programming. What we really wanted was a 701 but we couldn't get it.

We started looking around at the options. The 704 was too far downstream so we looked at the Univac 1101. Actually the 1101 and 1103 computers had been developed by Engineering Research Associates and later purchased by Remington-Rand, so at the time we were talking to Remington-Rand.

As this particular application required considerable use of tapes, tape reliability and tape speeds were very important. The tape drives on the 1103 were built by somebody else and just couldn't hack it, so we proposed to Remington Rand that they make a proposal to us for taking the tape drives from their own gear and putting them on the 1103. We said, L"Look, if you'll put Remington Rand tape drivers on the 1103 system, we'll buy one if you'll do it for a reasonable price." It wasn't that the price was outlandish, but the amount

of time it was going to take them was. About this time the Navy decided they were going to turn back their 701 and get a 702, so it looked like we could get a second 701. We told Remington Rand, "Sorry, your response was just inadequate." They wanted an incredible amount of time! They wanted something like two years in order to pull this marriage off.

I remember that visit very well because it was the first time I met Seymour Cray. He was just a programmer in a group at that time, but obviously he was the brightest guy in all of the ERA portion of Remington Rand.

MAPSTONE:

What I start to see happening, or that happened, was that the user caught on to the real significance of the computer before the people who were making it did; before the Univacs and the IBM engineers.

ARMER:

I think so, going clear back to the days when we were using punched card equipment. We discovered things you could do with the 602 that IBM never dreamed of. Things were done with the tabulator, a device which can't even multiply or divide, but we used to make rather elaborate tables by the use of differences. I guess this had been discovered by people working in astronomy long before we did, it wasn't anything that we thought of doing. It was a computational scheme which had been used in the production of the first tables of logarithms hundreds of years ago in France, but it had been mechanized on an IBM tabulator, doing things with the gear which IBM never dreamed were possible. At this point in time it was quite the case that the customers were much smarter than the manufacturers.

Another vignette: Before 704s were delivered, we concluded that we needed a lot more memory for our 704 than IBM was offering. We had already had the experience of being able to go to IBM and say, "Develop the CRT device for us," so we went to IBM and said, "We need a big memory."

IBM said, "Okay, but we don't think there's much of a market for that, and if we're going to do it we're going to have to have a development contract with you." In other words, you are going to have to pay for at least a share of the development costs.

We said, "Fine, we have a particular problem that we think we can't really do without the bigger memory." So the negotiations went on for awhile. This was during the time in which we had let a contract to International Telemeter to build the core memory for JOHNNIAC, and also at a time when IBM was saying that the early 704s would be delivered with Williams memories rather than with core memories. First of all, we had beat IBM over the head, saying, "Okay, we will just rent the computer, we won't rent your memories," and then we got North American, Douglas, and other people to say the same thing and to make it quite clear that we were willing to let contracts to International

Telemeter to build core memories for our 704s. As a result, IBM rescinded the announcement that the early 704s would have Williams tube memories, and said that indeed they would have core memories.

The core memories in something like the first eight machines were all hand-built. This response on IBM's part undoubtedly cost them a lot of money. The core memories were hand built almost in parallel with the prototypes, so when they discovered something was wrong they often had to not only change it on the prototype, but change it on eight other models. Then we said we wanted 32,000-word memories instead of 4,000 words.

IBM goes through periods in which they seem to be poor, and in fact I think it's a deliberate tactic of their management to decide, "Okay, this quarter we're going to shake the fat out of the organization, and we're really going to prune everybody's budget back."

MAPSTONE:

They still do it.

ARMER:

Yes. They were in one of these "feeling poor" periods when we were talking to them about the 32,000-word core storage. One Thursday, after the close of business in New York, Cuthbert Hurd called me and said that he would be in my office Monday morning with IBM's final proposal for the core store. I said, "What's it going to cost us?"

He said, "$500,000." I said, "At that price I'll take three."

At this point in time, the computer activities in the System Development Division were still under me, they hadn't been split off, so not only did I have my own 704 to worry about, which we already had, but SDD had two on order.

Cuthbert arrived in my office Monday morning, but on Friday, in New York, the decision had been turned around: IBM didn't want any money from RAND for development, they would prefer to lease the gear to us rather than sell it to us. It was a hell of a lot easier for me to get money to lease devices than to fight for a development contract, and although I had all the approvals to spend that amount of money, the red tape level was just an order of magnitude easier to lease than to buy. Further, since they announced that it would rent for $15,000 a month, I said, "Okay, but I still want three." It turns out again that by the time they got them into the field, they rented for $20,000.

Again, I think that maybe this was an instance in which our eagerness to go after a product changed IBM's mind about what the market was. I'm sure they thought tat the time, "Who needs that much memory?" In fact, Von Neumann, who had been a RAND consultant since the first days, had told my boss, John Williams, that he thought Armer

was wasting RAND's money getting IBM to build this big memory. He said, "Nobody needs a memory that big." He probably didn't, you know.

MAPSTONE:

He had such a good memory himself.

ARMER:

He had argued with me that one could always organize the problem in such a fashion that you could get by with a hierarchy of memories. He might have been able to do it; he was just that much of a genius. Although the cost of doing it is a trade-off between the machine costs and how much personnel time you can save by pushing some of that job off on the machine. Sure, we could have done the job with less storage, but it would have taken a lot more programming to do it, and further, a lot more machine time. I think that even with respect to the expenditure for hardware, we were money ahead by getting the larger memory. I think he was just plain wrong. But it might have been that since he had been a consultant for IBM, he was also telling IBM that there wasn't any market for 32,000-word core storage units.

MAPSTONE:

That's possible. Quite possible.

ARMER:

Also, another thing in that decision was that North American had gotten wind of the fact that we had these negotiations going on with IBM, and had sent a telegram saying they wanted one, too. Herb Grosch also found out, so he too was beating the drums that he wanted 32 K storage.

MAPSTONE:

Where was Grosch at this time?

ARMER:

He was at General Electric in Edendale, outside Cincinnati. Speaking of Grosch, I recall that meeting at which he sent the telegram to T. J. Watson. I remember one evening we went back to the lab and there was an organized tour. Grosch had once worked for IBM, and he said to me, "Let's ditch them, Armer, and find out what's really going on around here!" [laughter] He had a simple idea. When the lab had been built, no one had planned ahead to the times in which they would need lots of power for computers and their air-conditioning. When they rewired the lab to pipe around all this additional power, they had done overhead conduit. Herb had figured this out and he said, "All we have to do is

find that conduit and I bet we'll find a new machine." [laughter] In fact, we discovered the 703, which wasn't really a computer, it was a device built to sort and collate information on the tapes. It was a tape-sorter/collator. And we found a 705.

MAPSTONE:

[laughter] That was absolutely a dead secret, wasn't it?

ARMER:

Yes.

MAPSTONE:

Oh, fabulous, did you confront them with it, or did you just tippy-toe around looking at it?

ARMER:

No, we didn't. You learned, in some sense, to play games. You then get some IBMer that you think knows, and you start saying the little bit that you know, which is all you know, but you say it in the context that you know all about that.

MAPSTONE:

And then he confesses?

ARMER:

Then he confesses, yes.

MAPSTONE:

The FBI uses it all the time! [laughter]

ARMER:

It's amazing how that used to work. Indeed, we got some IBMer at that same meeting to tell us all about the 703.

MAPSTONE:

Oh, it's lovely. That brings us back in a way to what we talked about before the tape went on, and that was the position that RAND was in of having information that was privileged.

ARMER:

Privileged, as far as IBM was concerned.

MAPSTONE:

But also being very careful of this information.

ARMER:

In fact, I mentioned that somehow North American had gotten wind of the fact that we were dealing with IBM for a 32,000-word core storage. IBM was very upset about that, because they thought I had leaked this information to Jack Strong, so I had to go through a fair amount of trouble to find out exactly how the information had been leaked, and made sure that IBM then discovered that I was innocent. It turned out, as you might guess, that the IBM salesman had leaked it because in some sense he was just taking care of his customer. Herb Grosch had found out about it. I presume, in the same fashion, so our temporarily tarnished reputation for not being able to hold our tongues was restored.

MAPSTONE:

I'd like to talk about John Williams for a little bit. Because he's no longer with us, and you worked closely with him, his work and contribution.

ARMER:

John was head of the Mathematics Division at RAND. Although he was an astronomer by education, he didn't have a really strong formal educational background. To my knowledge, he never did any mathematics until the time he was at RAND, so certainly didn't make any contributions to the computer field in a technical sense.

His impact upon me was one of sort of a love of learning--it's very difficult for me to put it in words--and a sense of personal integrity. I don't mean he ever lectured me on personal integrity, it was just by the example of his own behavior. He was continually battling with Collbohn about research policy at RAND. John very much believed in the philosophy of looking for the very best people that he could find, hiring the guy and saying, "Okay, I just purchased from you your time for the next 3–5 years, here it is back, now go off and do with it what you think is best." He very much believed that people did their best when they were doing what they wanted to do. I think that much of RAND's success in the early days was due to this philosophy. It was also due to a lot of other things, but this is not a history of RAND and that's another day's worth of tape.

I remember back in 1968, when I had given notice at RAND, but was still around, I had given notice because there were a number of us who were planning to set up our own

company. That, however, was aborted, so I was looking for some place else to go. One of the places that I was thinking very seriously about was U. C. Irvine. Jule Feldman had written a memo to Aldrich all about Paul Armer and he wrote some things in it which I had not thought of before. He praised me for my ability to recognize good people, and good ideas, and to back them. I'd never thought of myself that way before, but it also struck me when I did think about it, that those were exactly John Williams' strengths.

I think that if one looks back at the contributions of Paul Armer to the computer field, they're mostly in terms of backing good people; backing Cliff Shaw; saying that I was going to give budget support to Newell, Simon and Shaw. They were good enough people that if I had said, "No," they'd have gone some place else and found some other way of doing it. Seeing problems, finding somebody to work on that problem, or convincing them that it was a problem they should work on. I think one of my more major contributions to the field is a book called The Economics of Computing by Bill Sharp. For five years I looked for somebody to write that book. It's really the only book on the economics of the computer field that I'm aware of. It's not quite the book I had hoped would be written, which still hasn't been written, but it's 75% of the book I was after. My only contribution was to see the need for it, to look for somebody, and then to say to them, "How about writing this book?" When somebody gets a reputation for writing novels, a publisher is quite willing to say, "Okay, here's a big advance, go off and write this book." The market for technical publications is not that flush, and publishers don't go around to people and say, "Here's $30,000, take a year-and-a-half and write that book." Due to the somewhat unusual circumstances at RAND, I was in a position to say that to Bill Sharp. The interesting thing is that the royalties which came back on the book, I believe, exceeded what was paid to Bill Sharp. That, in some sense, is the impact of John Williams on me.

MAPSTONE:

And therefore on RAND.

ARMER:

Oh, I think John Williams had an incredible impact on RAND. A similar, but I think much lesser, role was played by Charlie Hinch.

MAPSTONE:

I don't remember him.

ARMER:

He's the president of the University of California at … John was the head of the Math Division and Charlie was head of the Economics Division. The two of them were always fighting Collbohn about the liberal versus conservative policies with respect to

research. [pause] I think that it was John who had the greater courage of his convictions in these battles. John had hired Cecil Hastings to begin with, which really got RAND started in the computer business, because Cecil then said, "Okay, we need IBM punched-card equipment to do this." Then George Brown got on board and we got into the analog computer business. Then John and George and Bill Gunning made this trip and of that trip John was the senior guy as far as clout in the corporation was concerned. When they decided what RAND needed, it was John who convinced management that they should put up the money to build JOHNNIAC. John was also quite close to Von Neumann.

A vignette about John. I said I'd never known him to do any mathematics. Further, during the days of the 701 and even the punched card equipment and the early days of JOHNNIAC, he never used the machine in terms of writing a program himself, or even having some work done for him. When JOSS came out on JOHNNIAC, we put a console in John's office and he became a real addict. I remember one Labor Day weekend in which we were working on the specifications for a follow-up to JOSS, hardware-wise. We were going to buy a machine and eventually bought a PEB6. JOSS on JOHNNIAC, because JOHNNIAC was such a limited machine, could really handle eight consoles and nothing more, so we needed much more than that. We were sitting in the conference room working on these specs and John was apparently off in his office using JOSS. He came roaring into the conference room. Now this is a story not only about John, but also says something about what was built into JOSS. Earlier I mentioned the error messages. These messages were nested in terms of complexity, and there was an error message which you could get which said, "Things are so fouled-up, let's start over." [laughter] Well, John had been using JOSS for at least a year or so, but had never gotten that message before. He had gotten that message just then. In talking about JOSS II, we were now talking about decommissioning JOHNNIAC, and John comes roaring in with the output from this typewriter saying, "You can't kill that machine! It's alive! Look what it just said to me."

Which if I may, reminds me of another vignette. There was a guy in the Math Division by the name of Oliver Grosse who would use JOSS during its development period and try to find ways in which he could crash the system, or try and discover ways in which, say, he could get the system to send messages to Cliff. Somebody had the idea, I don't know who it was but I thought it was Cliff, that they were going to play a joke on Oliver. When you signed in on JOSS, the first thing you did was to type your initials. Cliff programmed JOSS to wait for somebody to sign in with the initials O.G. JOSS also had the capability that if while typing you made a mistake, you could backspace and type over, and so it would wait until somebody had made quite a few of these backspaces and retypes within a short period of time. Now Oliver was a lousy typist, so Cliff programmed it such that when the combination of these two things; namely O.G. and a lot of mistakes in a short period of time, it would interrupt and say, "Oliver, for God's sakes when are you going to learn to type?"

The funnier part of this, in terms of if you knew Oliver, was that he was determined to discover how Cliff had done this, so first of all he had to get the machine to do it again.

Cliff knew that Oliver's behavior was such that he would try and get the machine to do it again, so he had programmed it so that it would do it this once and never again. Oliver apparently spent a week or so trying to make it do the same thing again and it wouldn't. Finally he came in to Cliff and said, "I give up."

MAPSTONE:

[laughter] Oh, that's lovely.

ARMER:

Well, not very important history.

MAPSTONE:

No, but it tells something.

I'd like to go back to something you said which referred to yourself, and that is the letter in which you were described as someone who could back good ideas. In relation to computing developments and RAND, can you go into some of these? You mentioned a couple. I was wondering if you could enlarge.

ARMER:

I think it was mostly people. JOSS is probably the more important example; maybe that's just a subset of the whole Newell-Shaw-Simon area. The idea for SHARE wasn't mine. The idea, despite Blair Smith, for DCA wasn't mine. I just thought these were good ideas, and I happened to be in a position where I had some resources to back up these ideas. Some of the other important things that I think we did at Rand: I think we were the first people to program a selection test. In fact, the way that we got into that was the result of Cliff Shaw.

At the time we hired Cliff we were looking for about three people, and weren't at all happy with any of the applicants that we'd had. We decided that we had to hire at least one person, and that we'd wait until two others came along. We decided to take the best of the applicants we had at the moment, and that was Cliff Shaw. Some months later we said to ourselves--and this was Don Madden and me--"If we hadn't been desperate we wouldn't have hired Cliff, so we're not doing a very good job of identifying good people." We decided to talk to some of the psychologists around RAND about programmer selection.

MAPSTONE:

Okay, let's turn the tape.

[Tape 2, side 2]

ARMER:

Don and I went to a psychologist who quite literally went to a barrel of tests--he had just recently moved to RAND, and his tests had been shipped in a barrel--and gave us a large number of them. After we talked to him, he said, "I think maybe these might measure some of the things that you think are important." We looked these over, selected a fair number of tests, enough tests that it would take a day to take them all, gave all of them to a sample of programmers at RAND, at Douglas, and at Inyokern; again something that we could do because of the feeling of cooperativeness that existed in the area. We correlated the results of these tests with ratings by supervisors of how good these people were at programming, reduced this to a much smaller sample of tests--Thurstone's primary mental abilities tests and Thurstone's temperament test, I guess it was called--and used these tests for a long time in hiring people at RAND. In particular, most of the people that SDC hired in their early days were hired on the basis of those tests.

Whether or not these tests are any good is still a matter of some controversy. Much of the word that they're no good is coming from RAND, particularly in the form of Bob Reinstedt. They were just an exceedingly valuable tool to us, particularly in staffing for SDC. The trouble with this, as we were using it, is that all you can do is kind of give testimonials. As a result of those tests, we hired people who we would not have hired, and people that we were happy with. But who knows how many people we didn't hire but who would have been very good. We never made the test an overwhelming criterion. Jules Swartz, the guy responsible for Jovial, just put on a very unimpressive interview, but he did exceedingly well on the test, so we hired him and were very pleased with him.

I think we also did the first salary survey in the field. I was continually battling with our management about what I could pay my programmers. I decided it would help if I had some evidence that other people were paying their programmers more than we were paying ours, and conducted the salary survey using the SHARE membership. This was not a SHARE activity. But I called a meeting at one of the SHARE meetings and said, "Hey, I'd like to take a salary survey. Won't you cooperate?" And most of the people did. SDC subsequently took up this activity and ran salary surveys every year for quite a few years. Then the guy who was doing it went to work for some other company and I don't know whether they still do that or not. The salary surveys that I see published these days are done by recruiting firms, and not very well done.

Another thing which was sort of part and parcel of this selection business, I mentioned a few minutes ago Bob Reinstedt, a psychologist, who came to RAND as my administrative assistant. He'd previously been a Dean of Men at Whittier College. I was concerned about psychological aspects of computer people as a class, and got him interested in the problem. He eventually started something called the Computer Personnel Research Group, which eventually became a special interest group within the ACM.

That, I suppose in some sense, is another example of some of the contributions I think I've made to the field. The energy was his, the ideas were his, but I was not only in a position to back him with RAND resources, but to eventually go to ACM and say, "Look, these guys have an active group, but they're a different kind of organization in the sense that some of them, not even half, are computer types; some of them are personnel people and some are psychologists. They have their own organization at the moment, but because they're growing they think they should get more of an office, so to speak. I think that what they ought to do is become a special interest group within ACM. But if you're going to do that, you're going to have to change your rules about what's a special interest group within ACM, because these people who are psychologists are not interested in joining. They're not interested in paying $15.00 a year to get a bunch of journals they can't even read.. By dint of a fair battle, I convinced the ACM council that they ought to change their bylaws in order to take in a group which was made up of people other than entirely computer-niks. It's still a viable group, and I think has made some contributions to the field.

MAPSTONE:

How about education? Not just educate programmers, but just spreading the word through universities and learning institutions on the West Coast.

ARMER:

RAND probably wasn't too much involved in that sort of thing. Right off, we began to have trouble hiring anybody with any kind of experience. As a result, one semester Madden and Mehlan and I taught a class at USC. We soon decided that there wasn't enough of a multiplicative factor, so to speak; that it took up too much of our energies and produced too few students. Fred Gruenberger certainly played a role in that as a Randite, and in fact RAND, through Fred, did a fair amount in trying to interest secondary schools, and secondary school teachers in the computer field. This was particularly so once we got JOSS, which was easy for the youngsters to learn. After Fred left, Shirley Marx did some of this. We'd bring several summers, students and teachers from the Santa Monica and L.A. school districts. Again we were concerned with the multiplicative factor. So Fred was trying to get at the teachers rather than just a handful of students.

Another impact that RAND had on the field was through Fred Gruenberger. He had the idea that once very year, just before the fall joint computer conference, we should have a bull session of the leaders in the field. I said, "Fine," and backed him with some resources. Who knows what impact these had on the field? We sometimes used them for particular purposes. At one point in time, there was a movement within the Air Force to standardize on JOVIAL as a programming language. At that point in time we felt it was premature. To some extent they've since done that, but I still believe at that point in time it would have been a bad idea. So when inviting people that year we set up the agenda for what we were going to talk about. The idea was to present an argument against

standardization now. We produced a document from this meeting, and I think it slowed down that go-to-JOVIAL move.

Another thing is documented in this letter to Walter Carlson that I gave you. Back in 1959 I was vice-chairman of the National Joint Computer Committee, and there had been a fair amount of discussion in those days about somehow getting the various professional societies together. About this point in time, the International Federation of Information Processing Societies (IFIPS) was formed, and so it became a question of which American society was going to represent the U.S. As vice-chairman of the NJCC I got appointed chairman of a committee which I think was called the Long-Range Planning Committee, and was asked to look at that question and also to look at the broader question, what about the problems facing the industry as a whole rather than those that face just ACM or just IEEE? I decided the way I was going to pick other people's brains on this one was to put that squarely on the agenda for the RAND symposium that year, and invite people that I thought could contribute. And we did that.

That night I wrote out the committee report, had it typed the next day, the next night sold it to the other members of the committee, and then we presented that to the meeting of the NJCC. My recollection is that with minor modifications, the committee report was accepted. As I say in that letter, I'm not trying to put myself up as hero for this, because it was when I happened to be NJCC vice-chairman and then became sort of an ex officio to be chairman of the Long-Range Planning Committee. I then picked the brains of these other people through the RAND symposium, and went forward with that idea.

These symposia still go on, although since Fred Gruenberger and I have both left RAND, the last one just said, the 14th Annual Computer Symposium. [laughter]

There's something else that I'm sort of pleased with, or was pleased with at the time. Back in the 701--702 days, I began bitching to IBM about the lack of reliability in their tape transports. They came back and presented me with a study that essentially proved I was wrong. That made me mad, so I set out to show what was wrong with their study. What was wrong was the fact that they had studied 702 installations, which were the commercial business-type installations, and they used their tapes in a much different way than people who were in the scientific computing business. The way they would use tape was to say that they were about to write checks for payroll lists this week. They had a master tape from last week and they had an input tape of what things were supposed to be like this week, and they'd essentially run one tape against the other, arrive at a new master tape, and take all the tapes off and go on to the next job.

We, the scientific users, were using tape in the fashion of another level of storage. Say we were doing linear programming and we'd have the elements of a large matrix, we'd store that on tape. Consequently, we'd continually be rereading that back and forth, back and forth, rather than reading it from beginning to end, rewinding it, and taking it off, which was the usual commercial application.

Then I got Keith Uncapher to take microphotographs of the heads and of the tapes so we could prove to IBM that they were doing a lousy job on the production of their tape heads, and that they were like fingernails gouging the tape. This went on for a long time, and after the fact I went back and found out what we had spent convincing IBM that this was a problem and, further, what were the details of the problem and what they could do about it. I was flabbergasted to find out we had spent a quarter of a million dollars.

MAPSTONE:

How?

ARMER:

But, you know, on the other hand, I think it was worth one hell of a lot more to the industry. We showed IBM what was wrong with their tapes and got them to correct it. Again, being at RAND was kind of an unusual position, whereby our philosophy was if it's good for the field, or in a larger sense, if it's good for science then it's a reasonable expenditure of money. In those days there was no way that Douglas could have justified the expenditure of a quarter of a million dollars for improving tape reliability. If they still had that problem today, they undoubtedly could, because tape reliability could easily be costing them a quarter of a million dollars. But it wasn't in those days, and it wasn't costing RAND a quarter of a million dollars' worth of difficulty, but it was certainly causing the industry that kind of grief. So RAND was in a somewhat unusual position of being able to spend money to do things which were for the good of the industry and not to be very parochial and self-centered about asking what was a reasonable expenditure.

MAPSTONE:

I know you were very much interested in what the other companies were doing, what they were building, and you'd get reports on them. Did you use this information in any way except just to keep abreast of the art?

ARMER:

Not really. And to keep the Air Force informed, so that they were abreast of the state-of-the-art as well.

MAPSTONE:

Did you recommend to the Air Force, for instance, that maybe they should get an XYZ computer for such and such a need?

ARMER:

No, we were usually very careful to avoid ever doing that. I felt very badly in some sense over the years about the way in which we contributed to IBM's dominance of the market. We did that in a sense by coming up with applications which we developed on IBM gear and then turned over to the Air Force. Of course, the simplest way for them to pick up the application and use it was to use IBM hardware. In an indirect way, we sold lots of machines for IBM, I think IBM recognized that, and to some extent that is one of the reason, I think, that for years we were probably IBM's most favored customer, at least in the scientific computing area.

MAPSTONE:

And, as you just said, you also did some of their trouble-shooting for them in a sense, since you saw the problems, picked them up and went ahead.

ARMER:

Yes. I don't know whether these vignettes are a waste of time or not.

MAPSTONE:

No, they're not.

ARMER:

One time we visited North American at the rocket line facility and talked to an engineer by the name of Martin Kline. This was when we were working on this tape reliability problem. He said, "I'll tell you how we solved that with respect to some of the audio tapes that we have around here. We use something called malidium disulfide…" I guess it is. I'm not sure that's quite right. Let's call it mally. Anyone who knows lubricants will know what it is, because it's a somewhat standard dry lubricant, like graphite, but with different properties. So, he says, "In fact, I'll give you some of it to take home, but be careful if you put it in your pocket, because by the time you get home it'll be out of the container and into your pocket." Which turned out to be quite right.

Anyway, we did some experiments with this at RAND and it seemed to make quite a bit of difference. We immediately took some of it back to Poughkeepsie to get IBM to try this. Well, the vignette is that shortly thereafter, Vin Learson, who was quite a yacht-racing enthusiast, had the model basin at MIT doing experiments with models and boats with malidnium disulfate coated on the hulls. [laughter] Somehow the technology got to boats faster than it did to the tapes. It turned out that this property of the mally getting into everything was somewhat of a disadvantage and IBM discovered other ways to make their heads smooth so that they didn't need the lubricants.

The problem was solved in another way, but it always struck me as sort of humorous that we got action much faster out of IBM with respect to Learson's boats than we did with respect to the tape problem.

MAPSTONE:

Another thought which has occurred to me while we were talking is that RAND really was a futures company. Were you, the people at RAND, conscious of the future of computers in any way; did you see them as the industry of the future?

ARMER:

Really I think we lacked that kind of vision. I was telling you earlier about how we went to IBM and said we've got to have larger memories. At that point in time, in some sense, we did see the need. I can remember much earlier on the first Card Program Med Calculators, you could put on one storage unit, called an icebox, which held 16 numbers, and then you could put up to three of these on so you could have 48 numbers. Original plans for the JOHNNIAC were, i guess, for 4096 or 2048 words. Because of the expense of Selectrons we only had 128-word storage on JOHNNIAC when it had a Selectron memory. About this time the people at Las Alamos were talking about Maniac II which was going to have a 10,000-word storage. Like Von Neumann I said, "My God, who needs 10,000 words of storage."

Certainly none of us at RAND got rich by saying to ourselves, "One just needs to put one's money into computer stock." When IBM eventually decided that they were going to build the 701, I gather Cuthbert Hurd was having a hell of a time convincing them that they should build seventeen. There were other people saying, "Half a dozen of these machines will be all that the country can use." None of us were saying, "You're out of your head, it can take a lot more than that." I don't think we were explicitly asking ourselves, "What is the market for these things?" and agreeing that it was that small. We just didn't think about it.

By the time IBM was making market forecasts for the 650. I remember telling the IBMers who were making this forecast that they were being extremely conservative; indeed they were. By then I suppose we recognized that it was going to be a much larger and a much more important field than we had early on. In the early fifties we were just so damned busy doing it that we weren't much thinking about the wider world. In fact, until like 1962, when I started thinking about the social consequences, I hardly thought about any of these things. It's not that I had dismissed them, they just never entered my head.

Although, in retrospect, an interesting thing about the salary survey that I had conducted back in around 1956--I'm not sure of that date--is that I had been quite concerned with the problem of privacy, and had hired a public accounting firm to be an intermediary. We had each of the companies participating in the survey prepare a card for each of their people which had his salary, his educational attainment, how long he had been in the

field, his age and also some quoting of geography. They'd prepare a card for each individual and then ship these cards off to this public accounting firm, who then gathered them all in a box and sent them to us. The notion was that I could learn nothing from postmarks. In some sense, in retrospect, I was impressed with the fact that I was concerned about privacy at that point in time, although the word privacy never occurred to me then. An interesting aspect about that was when I got the box of cards, the first thing that I did was to sort the cards with the highest salary. I picked up these cards and I looked at the salary and I could also look at the coding which would say this was in the Pacific south, this was in the Pacific area, this was New England, and this was the South, this was the northern Midwest or something of that sort. I went through the first dozen cards, and could immediately say whose they were. That was also a lesson about the questions of privacy and most people who talk about that problem today still don't understand that just by taking a person's name out, you do nothing for the privacy problem provided there's enough other information.

Another example of the way I've contributed to the field is to sometimes go see good papers and then work like hell to get that paper published. Some years back Lance Hoffman and Bill Miller, at Stanford, published a paper on just this thing. Immediately I saw it, I said, "Gee, we've got to get this published in Datamation." Another thing that occurred due to happenstance, was that Datamation started up in Los Angeles, so every editor from the beginning has always been someone I've had access to. I've been able to say, "Here's something that's happening in the field that you ought to write an editorial about, try and get it turned around," or "Here's an important paper you ought to publish."

Many years ago, as I mentioned in this letter to Carlson, prior to the time that AFIPS came into being, there were a lot of people concerned about the fact that we needed to get the computer societies together. Herb Grosch and I went to a meeting at DPMA and said, "Look, we at least should be talking to each other." We got the DPMA people invited to a National Joint Computer Committee meeting at which they were really treated shamefully, mostly by the IEEE people, but also by some other ACM people, who essentially said, "You're nothing but EAM installation managers, you're really beneath us and we're involved in science. Go away."

MAPSTONE:

Oh, my God.

ARMER:

Really treated them shamefully. As a result, DPMA people have been rather skittish about AFIPS ever since.

MAPSTONE:

We're almost to the end of the tape, so let's just do a recap. That is, pick out the key important and significant developments--I think hardware is pretty obvious--in the field of computers as far as hardware, software, and maybe people themselves, are concerned. We've obviously talked about some of them.

ARMER:

The key ones in the field?

MAPSTONE:

Yes.

ARMER:

Or try and emphasize on the ones which RAND had some sort of a role in?

MAPSTONE:

Let's say both at RAND, and if you have any feelings about the field in general. I mean, if you felt that such and such a programming language was a great breakthrough, even though RAND was not involved in it. This, of course, is opinion only.

ARMER:

In hardware, certainly the original concept of the stored program. As we were discussing in L.A., who does it belong to? Is it really Von Neumann's? As IBM or as Eames might take it to be. That's the idea of ideas in this whole thing. The next thing which has had the most fantastic impact is the transistor. I remember writing to Forrest of Datamation recently that I thought it was incredible that they had devoted an inch to the fact that it was the 25th anniversary of the invention of the transistor.

When I talk about the social implications of computing, I like to use as a prop Hewlett-Packard's little electronic calculator. I pull it out of my pocket and say that when I entered the field, the amount of hardware represented here would have cost a million dollars, and would have taken a room to house it. In some sense the room is a red herring, because it's what's happened to costs. The essence of the social implications of computing is that it has just made a number of things exceedingly cheap. The reason we have a privacy problem is that you can now keep tabs on a larger number of people because it's very cheap to do so. We could have done it before, but it just cost too damned much money, so you didn't do it.

There's an analogy, due to Hamming that I use all the time, in which he points out that we can walk around 4 miles an hour, we can get around in cars, if there isn't too much traffic, at 40 miles an hour, and we can get around in airplanes something like 400 miles an hour.

Each of these differs from the other mode by the factor of ten, or an order of magnitude. Ignoring space travel, going from 40 miles an hour to 400 miles an hour took something like 50 years; one order of magnitude in 50 years. In the computing field we go through 3 or 4 orders of magnitude every decade, and in particular that is still happening in economics.

B. O Evans--I'll tell you a vignette about Bob--recently gave a speech at MIT, the title of which was, "Revolution, Not Evolution," about the economics. He was essentially looking downstream. Obviously, these changes of 3 or 4 orders of magnitude in a decade has got to stop some time, but for at least another decade into the future, they just don't seem to be leveling off in terms of cost. Hamming's analogy is, what if you were to say that 10 years from now automobiles were going to cost a thousandth of what they cost today? Or that a house was going to cost a thousandth of what it costs today? That's obviously going to have fantastic social implications. But that's what's happened.

How did I get started on this one?

MAPSTONE:

Economics?

ARMER:

Oh, we were trying to summarize, weren't we?

MAPSTONE:

Yes.

ARMER:

The key thing on the hardware side was originally the transistor, which not only made things a hell of a lot cheaper but made things a hell of a lot more reliable, and all the subsequent developments that continue to lower the price and increase the reliability.

Breakthroughs in the software area? First of all, going to assembly programs and then to compilers. In that sense, FORTRAN represented a major discontinuity in the field. Up until that point in time it really took a specialist to get a programming application going on in a machine, whereas once FORTRAN, and things like it, were around, engineers began to do their own programming. In fact there came a time, at least in the aerospace industry, where a large number of the people that they had hired as FORTRAN programmers became technologically obsolete and in surplus because the FORTRAN programming was taken up by the engineers who actually wanted the work done, and they didn't need the middleman. So that really increased the demand for computing because it meant a much larger set of people could use them.

[End of Interview]