



Smithsonian
National Museum of American History
Lemelson Center for the Study of Invention and Innovation

Computer Oral History Collection, 1969-1973, 1977

Interviewee: Franz L. Alt

Interviewer: Uta C. Merzbach

Date: March 13, 1969

Repository: Archives Center, National Museum of American History

MERZBACH:

Well, let's start talking about the relationship of your early training or experience to your later work. Could you tell, in a bit more detail, what your mathematical specialization was originally, as a student in your professional career, before you became involved with computers?

ALT:

There was no intentional connection, of course. When I studied mathematics, I didn't think that I would get into computing. When people were looking for personnel to put onto computers, they didn't particularly look at my early training; they just wanted a mathematician of some kind, but it turns out that the connection was better than might have been hoped, and also better than most people would acknowledge today.

When I studied mathematics in Vienna, the emphasis was on quite abstract subjects: on foundations of mathematics and mathematical logic; on fields like graph theory and combinatorial topology, and things like that. That was the atmosphere at that time, and these were the interesting subjects. Schlick was there in logic and philosophy, and Carnap was there. Goedel, whom we now call the foremost logician, was a student in Vienna, and he was a few years older than I. That was a sort of thing in which mathematicians in Vienna were interested, and I have always felt that these areas have a stronger bearing on working with computers than so-called applied mathematics. I had almost no background in applied mathematics. I never even took a course in differential equations, I believe. I have always found that it is the theoretical mathematicians, even number theorists perhaps, who make the best computer people. Sometimes people ask why one has to be a mathematician at all to work with computers. One doesn't have to be; very few of the things that we learn explicitly in mathematics have any bearing on computers. It's more the general attitude toward problems that carries over. A mathematician is in the habit of looking at a problem and solving it in the most economical way. That means with the smallest number of assumptions, with the fewest tools that one can possibly get away with. One is not so much interested in proving a theorem, but rather proving it from the weakest assumptions. There's just as much credit in pointing out that one assumption is superfluous, or was implicit in the others; there is just as much credit in that, as there is in finding a whole new proof for something.

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

And this is so characteristic of work with computers. We write a computer program and it would be important to recognize that it applies to a great many situations which look very different; that the same programs can be applied, perhaps, to bookkeeping and pre-information retriever, or things like that. One comes to that realization by formulating a problem abstractly, forgetting about the meaning of the numbers that go in, the data that goes in, and thinking only of their structure, of the logical relations. This is what mathematicians are taught to do, especially the highly theoretical mathematicians.

I have always felt that people who look for applied mathematicians in order to use them as computer programmers, are going in the wrong direction. It is true that there are good programmers without any sort of mathematical training. That is possible, and one would be inclined to think that at least they would have made good mathematicians if they had tried to study mathematics. But we did have the explicit experience that when we hired new people, those with vast mathematical training were the fastest in learning how to program for computers. Quite typically, somebody with a Ph.D. in mathematics would usually be a good programmer in a week or two, and somebody with only a bachelor's degree would be a good programmer in a couple of months, perhaps, and somebody without any training in mathematics would take considerably longer, typically. There are wide deviations from average, but I think one can generalize to that extent. So, from reality with mathematical logic foundations, systems of axioms, topology, abstract algebra, number theory, all these things helped in getting used to computers, getting accustomed to computers. There were other areas. For instance, I did have a lot of complex variables, at least four semesters, but that didn't help me particularly in working with computers; that helps only when you happen to program a problem in that field.

MERZBACH:

Yes. What about any other related subjects, in your own experience, outside of mathematics? Would you say there was anything else that was relevant, or perhaps anything that you did not have in your own background that you felt would have been useful, in addition to this training in pure mathematics?

ALT:

Oh, there were lots of things that I did not have that might have been useful. I knew nothing at all about electronic circuits, for example. It would've been nice to know, but none of us can have enough education. There were lots of things in my education that had no bearing on computers and that were wasted. What I wanted to say was that just those things which some people might have considered wasted, were not; namely, the highly theoretical mathematics. But in between, I had spent time in life insurance mathematics, and economics and statistical business forecasting, and that was not particularly helpful in working with computers. One thing that was, perhaps, was that I had some experience in organizing manual computing. I had a group of a half-dozen girls or so, doing computing for the purpose of business forecasting and the experience in laying out worksheets, and assigning work to them and checking, and especially checking, was quite

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

useful. Other people have made this comparison, too; programming for a computer is something like laying out the work for a hundred, or so, girls with desk computers.

I had handled desk computers myself, and I was quite expert with them, but that didn't help very much. One might say that I had become aware of the importance of rounding errors and things like that, but that could have been learned without all the experience that I had in practice.

MERZBACH:

Aside from that, did you have any experience with mathematical machines of any kind?

ALT:

No, I had not even heard of them. Maybe I had one course in college, and it didn't strike me as very interesting, then. I had to use desk computers in some of my jobs, but I didn't enjoy them particularly, and nothing of that carried over into computers.

MERZBACH:

You mentioned last time, I believe, that you had no experience with punch-card machines.

ALT:

I learned punch-card machines on the spot, and it was easy enough to learn them. I don't believe that experience with punch-card machines would carry over into computers, except for the administrative experience, in the same way as one profits by organizing a computation for manual computers, so one learns to organize a computation for punch-card machines.

MERZBACH:

Did you have any particular interest in mechanical devices, even as a boy; any particular gadget?

ALT:

No, no. On the contrary, I stayed away from them. I found that they were rather too simple-minded to be interested in. They were nothing compared to the complexity of a mathematical problem.

MERZBACH:

Yes. What would you say made you change your mind?

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

ALT:

Nothing has made me change my mind. I still think that there is not great interest in mechanical gadgets. I like to play around with them now, and repair them, but that's as a pastime. That's an easy way to relax and not because it's particularly interesting. One thing in this connection also, is that electrical engineering and building electrical circuits has always struck me as much easier than anything mechanical. Mechanical design is really an art, and requires a great deal of experience, and requires a great deal of mathematical insight of a kind that I don't have--geometry. By comparison, I have the impression that designing electrical circuits seems to be a much simpler affair.

MERZBACH:

Also, if one makes that division, you would consider yourself more algebraically oriented than geometrically, wouldn't you? Is that fair?

ALT:

Well, I don't know; I'm not sure there. I've had a lot to do with geometry, but again it's a simpler and easier thing compared to very abstract mathematics. I, myself, like to visualize mathematical problems geometrically, because that's easier to do, because that makes them easier for myself. It's a heuristic way, but what carries over to computers is the abstract formulation.

MERZBACH:

Yes, yes. Well, that's interesting.

ALT:

In other words, if I think geometrically about something, I'm avoiding the real problem; I'm using a crutch; I'm trying to make it easier for myself. The real problem has been too hard for me, so I'm trying to solve an easier problem instead; namely, the geometrical visualization of the problem.

MERZBACH:

Uh-huh. That's interesting. Well, perhaps we should turn to something else, unless you have another comment.

ALT:

No other comment except that many of the things I've just said were quite unpremeditated. They may turn out wrong, so the reflection...

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

MERZBACH:

Right. Did you have any comments on your Army career? I know you indicated you didn't think that it really was anything more than accident that you did manage to end up at Aberdeen.

ALT:

It was quite accidental and good fortune; it was very lucky.

MERZBACH:

What had been your previous assignment?

ALT:

Oh, I was in the ski troops, because I knew skiing. In fact, I think I must have volunteered for them. I was with the so-called light-mountain infantry, and they were never used as ski troops, as far as I know, but I was with them for about a year. Then that division was transferred to Texas for further training; I didn't like it very much down there, and one day an announcement was made that we could volunteer for officer's training, and so I jumped at that just to get out of that organization. Since we were no longer in the mountains, I was no longer interested in it. That was probably the wrong information, but you know that in the Army, one never knows where one is. I believe the division was, shortly after that, transferred to Italy and was used as a mountain-fighting outfit, and I might have liked it there. Meanwhile, I was in officer's training, in Chemical Warfare Service of all things, because that happened to be an opening, which was announced, for which I applied, and they felt that having had a minor in chemistry and about three courses at the university qualified me for the Chemical Warfare Service. In fact, after going through that, I was used for a short time as a statistician on some chemical experiments, but that was only a matter of weeks, I think.

Meanwhile, the Office of Strategic Services got hold of me because they thought that with all of my experience, languages was the most important thing for the Army. So, they pulled me out of Chemical Warfare Service and wanted to transfer me to Europe, I believe, for some kind of Intelligence activity, but the war in Europe ended before they could do that. By this time, I suggested that I be transferred to Aberdeen because I'd heard about it on and off, and I thought that really the most significant thing that I had to offer was mathematics, and the Army had not used any of that up to that point, and it was time that they did. I only knew that mathematics was used at Aberdeen for firing tables; I knew nothing about computers.

MERZBACH:

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

Well, when you did get to Aberdeen, could you describe a little bit of the administrative set-up at Aberdeen, how it was organized, and how the civilian and military units interlocked.

ALT:

I can give you my impression of it, though I'm not sure that that's really right. I didn't have all that much opportunity to observe, and it may be a one-sided, biased affair.

I was in uniform, and I was assigned to an organization which was heavily civilian; most of the people there were civilians. On reflection, I must have belonged to some military unit, but I don't even know what it was. It was probably a company or something of which I was a member, but I wouldn't know. I don't remember who my superior officer was, to whom I reported. For all I know I never saw him, although that's probably wrong. In practice, I worked for a civilian superior, as if I had been a civilian employee, with very few exceptions like, once in a while, I would be called out for guard duty, or something like that, a very, very small amount of interference like that, or be told to attend an indoctrination meeting, or a film, or something like that. For the most part, the military organization did not interfere with our main activity, which was civilian-scientific. The man to whom I really reported was Dr. Dederick, who was a civilian, and his superior was Colonel Simon, the Director of the Laboratories, who was a military man, but very understanding of the whole situation. He should get credit for really operating that organization very well and in a good scientific spirit. He should also get a lot of credit for, and I think he has gotten a lot of credit for, promoting the development of computers. He had to make all the decisions about assigning money to it, and so on; he recognized that there was something to look forward to. Colonel Leslie Simon. He became a General later, and was, I think, Chief of Ordnance Research and Development for a time, before he retired. That's probably all I can think of about Aberdeen. There was quite a difference between Aberdeen during the war, and Aberdeen in 1947 and 1948, when I returned. In the first place, during the war, everyone was temporary and considered himself temporary, especially seeing that I got there only in 1945, just about the time the war in Europe had ended, a few weeks before VE Day. We all knew that the war in Asia wouldn't go on much longer and we were all there just as guests. I now think of my closest collaborators, Derrick Lehmer and Haskell Curry, who only arrived in the summer of 1945. They stayed for a year after that, because they had been hired for a year, and wouldn't have anything else to go to, but they felt that this was entirely temporary for them.

I found that many of the civilian scientists complained about military regimentation, but I think unfairly. I think, especially, the Ballistic Research Laboratories at Aberdeen were operated quite liberally, and could almost stand comparison with industrial research laboratories, but not with universities, of course. Later on in 1947 and 1948, there was a much smaller group of people, but they considered themselves more permanent. They were residents of Aberdeen, and they formed more lasting friendships, and they were more interested in extracurricular activities. We organized seminars, and we organized

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

theater and music evenings, and things like that, and we were interested in the schools. At both times, and perhaps even more strongly the second time, we felt a cleavage between the laboratories and the personnel of the laboratories, and the local people of Aberdeen. The local old-timers always considered almost everyone who worked for the Proving Ground as an intruder, and this was felt more keenly in 1947 and 1948, when we considered ourselves residents. We were considered as outsiders by the local people, and we felt that we were being exploited and overcharged, and things like that.

MERZBACH:

How large a mathematical group was there--specifically, mathematical group?

ALT:

Somebody else would remember better than I; also the question is not well defined, and also, the mathematicians didn't particularly stick together. I was as close to some physical chemists as I was to some of the mathematicians. In the organization, there was a computing laboratory, and I suspect that there were about two hundred people at its greatest size, which was during the war. My recollection is that it was less than a hundred when I was there the second time, and had more to do with it; the first time I was outside the Computing Laboratory. Albert Bennett was the Chief of the Computing Laboratory, the Division Chief, he is the mathematician from Brown University. The computers were kept out of his cognizance, out of his competence, because it was felt that he ought to see to it that the job of computing firing tables and bombing tables was running smoothly, and he should not be bothered with extraneous matters. This is why, during the war, the Computations Committee was set up outside of the Computing Laboratory, directly reporting to Dederick. Of course, the few of us on the Committee had close contact with the Computing Laboratory; we were housed next to them and we tried to study in detail their operations, and we had good cooperation from them, but it was a separate organization. After the war, when I came back for a second time, the computers were in the Computing Laboratory and they were a very important part of it. By that time, I was Deputy Chief of the Laboratory and Dr. Dederick himself acted as Chief, Bennett having gone to Brown. We had two kinds of sections in the Computing Laboratory: there was one group of sections that were organized--the long problems, the Firing Table Section, the Bombing Table Section, the Data Reduction Section. There was another group of sections that were organized according to the equipment which they operated: the Differential Analyzer Section, the Punch-Card Machine Section, the Bell Relay Computer Section, and the ENIAC Section. Now, any problem might have belonged to two of these depending on the type of problem and the type of equipment, but that didn't seem to create any difficulty; it was, in fact, a good set-up, I believe, it could be imitated. As I said, there were, perhaps, close to a hundred people in the Computing Laboratory at that time, in 1947 and 1948, but that's not the same as saying there were a hundred mathematicians. Many people in the Computing Laboratory were hardly mathematicians; they were perhaps hand computers. And there were mathematicians elsewhere, particularly in the Exterior Ballistics' Laboratory and a few very good ones in Interior

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

Ballistics, some of my best friends were there. At that time, in 1947 and 1948, we organized a kind of seminar on mathematics and mathematical physics. There were perhaps twelve members, with only three of them from the Computing Laboratory, and the others from Exterior Ballistics and Interior Ballistics, and so on. You might say that these twelve were, perhaps, the core of the mathematically interested people at Aberdeen.

MERZBACH:

We wanted to get a few details on relay computers, and perhaps you'd like to take it in the order that seems most convenient to you. I think you indicated that there was not a great deal to be said about the IBM relay.

ALT:

No, they were small machines. For IBM, they were a long step forward from their conventional machines, but they didn't contribute much to the art of computing. They were programmed by plug boards just as conventional machines were, and they used the same components, and had about the same speed. In fact, we found it hard to find problems which would perform much better on the relay calculators than on conventional punch-card machines. They were a step-and-away for IBM toward their 604, and the 604 was a machine that was of importance because of the large numbers in which it was produced, but it was not of great importance to the art of computing. It became an important tool of computing for a time, and I guess the main importance of the relay calculators was that it taught IBM how to design the 604. On the other hand, the Bell relay computers were a real advance on the art in many ways, much of which got lost, as a matter of fact. They were not terribly influential, not as much as they deserved to be, and that was the fault of the people who designed the electronic computers later on, and didn't have the patience or wisdom to look, in detail, at the design of the Bell relay computers. By this time, perhaps, electronic calculators have caught up with all the fine points that were put into relay computers, even at that time, but it has taken a long time. The logic of the Bell relay computers was far ahead of its time and really admirable. For some reason, the people who designed electronic computers later on were preoccupied with speed, and the design of components, and the size of memory, and they didn't have time to look at the fine points of logic design. But it is true that the relay computers had little influence on the further development of computing; they might have had a little more, but they had little. It is also true that the electronic computers deserved their preeminent role, because the advances in logical design of computers would have been quite useless without the speed of electronic computing. That is, the Bell relay calculators were "white elephants"; they were maldesigned, or you might say they were over-designed. Something as slow as a relay calculator doesn't need all the fine points that the Bell Laboratories put into them.

MERZBACH:

For example?

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

ALT:

For example: unattended operation for many hours; for example: an enormous amount of self-checking. That was the outstanding feature of those machines, it was just impossible to get a wrong answer out of them. There were many occasions where the machines stopped, because one of the hundreds of internal checks would not come out right, usually because the checking circuit failed; the computation was right anyway, in the first place. All this was overdone for something as slow as relay calculators, because they didn't produce all that much. It would have been fairly possible to do all the checking manually, but it would have been of great advantage to electronic computers where, even to this day, there is not as much checking as we had on those machines. But it is true that all these advances would have been unnecessary without electronic speeds. On the other hand, electronic speeds were of great advantage even in the primitive way in which it was actually used. We might have gotten ahead a little faster, perhaps, if the fine points of the relay calculators had been better known and better observed by the designers of electronic machines. The Bell Laboratories computers were not the only ones in that way. Aiken, at Harvard, also advanced the art of logical design in slow computations, but I think in these respects, he was outdone by Bell Laboratories. I don't know what else one could say about the relay calculators, apart from their reliability, and high degree of automatic operation. People don't realize, perhaps, that it was possible to put on five problems on one of these machines, and the machine would automatically do one after another, check to make sure that the right data tapes were loaded for each, something that nowadays, no computer does automatically; we usually put that into the software these days. In fact, you might say that the Bell Laboratories computer had, wired into the hardware, most of the features that we now put into the operating systems of computers, and it has taken us twenty years to put them into the operating system.

MERZBACH:

You indicated last time, that there was some exchange. Was there anything especially significant about interchange of ideas among, say, the Harvard group, and those who worked with the Bell calculators?

ALT:

Yes, but I know that only from recollection, and I'm a little hazy on it. I do remember that Howard Aiken visited Bell Laboratories while I was there, from November 1946 to, I think, April or May 1947. I was on the Aberdeen payroll, but I was on detail service to New York at Bell Laboratories. During that time, I remember visits from Howard Aiken, and I remember that he was very cordially received, and I listened in to the conversations between him and the Bell Laboratories' engineers, and I had the feeling that there was no withholding of information on either side. That was the time when Howard Aiken was designing Mark II, and that means he made the transition from reliance on IBM components to reliance on relays, and more generally, the kind of components that

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

telephone systems had used before--punched paper tape, for example, in preference to punched cards. Well, he did use punched paper tapes even in Mark I, but it was a different sort of thing; it was relatively clumsy and he had to design his own readers. He also told us some things that were very interesting. I believe he talked at that time about magnetic storage, magnetic disc storage, the sort of thing that later developed into the magnetic drums of Mark III, I believe.

I remember something that I've never been able to verify. He had an ingenious number theoretic way of finding a location, a storage location, making use of the decomposition of an integer into its prime factors. I don't remember how it worked, but he told us about it and I was very much impressed with it, and apparently it was not put into effect. There were some things on which the Bell relay calculators failed completely. For instance, one did not recognize the need for storage, and neither did the first electronic computers. At that time, one thought that one needed to store about twenty numbers, and this experience came from the way worksheets for manual computers are designed. One generally doesn't work with more than twenty columns at a time, or, it is analogous to differential analyzers where one usually does not have more than, oh, a dozen, or at most, twenty integrators at a time, and each integrator can be used to store a function if there isn't any other storage but that. It was for these reasons that people felt, somehow instinctively, that maybe twenty numbers or so, was a good volume of storage. Von Neumann was the one who got beyond that and talked of storage on the order of a thousand numbers, a thousand words. It was Von Neumann who invented the concept of a computer word as representing either a number or an instruction. But, he also recognized that with electronic computers, we would run problems of much greater magnitude than before that would require first of all, storage of a different order of magnitude.

MERZBACH:

Going back once again, to the relay machines; was there any awareness of the work done by Zuse, in Germany?

ALT:

Yes, I read a report on his computer in 1945, before I left Aberdeen for the first time. I was not impressed; I thought that it was quite primitive compared to what we were doing. He made a lot of noise about the binary system which was rather a trivial thing to me, and to most of us. If we didn't design computers in the binary system, it was not because we didn't think of it, or because we couldn't have, but because we felt it unnecessary. It wasn't that much more complicated to make than decimal, and the saving was minimal and wasn't all that important. It seemed important for us to have computers work in practice, and I would feel that way today, too. Von Neumann made a switch to binary computers, and I'm grateful to him for that. I think it introduced an unnecessary complication; it was necessary to save those thirty percent, or whatever it saves, of electronic circuitry. It would have been easier from the start, to do everything decimally, to waste a few circuits, and make the thing conceptually easier to the large public. A good

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

many people were scared away from computers by the binary system. It didn't do any permanent harm, I suppose, and we have finally arrived at the stage where the inside of the computer is always binary, and all contacts with the outside world are either decimal or beyond that, alphabetic. We finally arrived at that stage, but it was unnecessary to take the detour of entirely binary machines.

Beyond that, I never saw that Zuse had anything very much to offer that wasn't done better, faster, and more efficiently, in the United States. And, of course, there was no influence of his work on the design of computers. They had already gone far beyond his designs. We got reports on him at a time when we realized that we were well beyond that.

MERZBACH:

When the first reports actually came out, was it before the end of the war?

ALT:

No, no; they were brought back by American Intelligence after the occupation of Germany. I saw them in, probably, August or September of 1945, and that was after the American Intelligence people had moved in. We had a good many visitors from Europe, not in 1945, but in 1947 and 1948; Swiss and Swedes came to learn from us, and of course, English people. It is not always realized that the British were ahead of us, ahead of the Americans, for a time. Of course, ENIAC was the first electronic computer, but it was not a computer by today's standards. We do not call something a computer nowadays, unless it has a stored program, and the first stored program computer was in England. The first working stored program was at Cambridge, built by Maurice Wilkes. It antedates the first American computer by a year, and most people have forgotten this fact, conveniently. The British also solved the problem of large storage faster than we did. Everyone here was dissatisfied with the early acoustical storage devices, although Wilkes used that successfully, and we were looking toward electronic tubes, a kind of television tubes, for storage, and we failed for a long time, after the British had succeeded. That was because their design was more conservative than ours; we tried to pack more into one tube, and failed for that reason. We also tried to make the electronic computer faster. Wilkes built his slower by a factor of ten, and succeeded where we failed, for a long time.

MERZBACH:

Did the visitors whom you mentioned, did any of them stay for any length of time? Well now, for example, I know that one or two of the people from Sweden spent about a term or so at Harvard. Did any of the foreign visitors spend any lengthy periods with you?

ALT:

I have to confess that I don't remember that, although I should have known about it because I was there, but I don't remember. Nobody that I can think of stayed officially,

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

was assigned to the Computing Laboratory, for more than a few days at a time, and yet, I know that... Oh, I know what I missed; if they did, they would've been in Philadelphia, and they would've been there during the time that I was away from the Proving Ground. In Philadelphia, for example, there was this famous course on the design of electronic computers, in the summer of 1946, I believe, and that's where people would've been and gotten most of their information.

MERZBACH:

Well, going back over some of these last items that you mentioned; what significance was attached to the floating decimal?

ALT:

Oh, that's right, I forgot to say. The floating decimal point was something that was implemented on the Bell relay computers, I think for the first time, and you might say that was something that was a lasting influence in computing. It came from there; nobody else had thought of it, and it was taken over. The first electronic computers didn't have it, because engineers had enough trouble without it. It was in fact, Von Neumann's idea that it shouldn't be done. It was Von Neumann's idea that the engineering should be as simple as possible, and everything should be done through programming. This is why the machine was binary and why it was fixed point, because he felt that anything like conversion to decimal and floating point could be handled by programming. But it is true that the concept of floating point was built into the Bell Laboratories machines, and I don't know whether that was worth doing or not. There was a long argument, for a time, of whether it was worth doing or not, and I'm not sure which way the decision should've gone, whether it was better to have it in the hardware, or do it by programming. Checking, I'm now sure, must be built into the hardware; one cannot possibly do all the necessary checking by programming. To do that would slow down the operations by a very large factor. Practically every single operation, or a very small group of operations would have to be checked, and that would lose a factor of at least two or three, if not more, in speed. It is well worth building that into the hardware just because it is performed every step of the way. That is not to say that one doesn't have to build additional checks into the programming.

The kind of automatic and unattended operation though, that the Bell computers did, got lost for a long time. We could put a number of problems on at the same time, and the machine would perform one problem after another. There were provisions for designating priority, a problem would be handled to completion or until it ran into trouble. It might run into trouble because a wrong data tape had been provided, or because there was inconsistency in the programming; the program, for example, called on a storage that wasn't there, that was occupied at the time, or it might run into trouble just because of malfunction of the machine. If for any of these reasons an operation couldn't be performed, there was a timer, and the machine might try again or wait for a while for the right information to come up. But after a matter of minutes, which was very short by

Computer Oral History Collection, 1969-1973, 1977

Franz Alt Interview, March 13, 1969, Archives Center, National Museum of American History

those standards, the next problem would be picked up quite automatically, up to, I think, five or six. There were provisions for subroutines, and the whole concept of a subroutine originated there, and it had a physical implementation there. The word "routine" originated with Bell Laboratories, as did many other words in the computer vocabulary. The subroutines had a physical implementation; all programs were put on paper tapes that were glued into loops, that was the only way. It was then realized that most programs were repetitive--oh, I should say all but one--we knew enough about the design of problems then to know that every computing problem has one routine that runs from beginning to end without repeating, and that was called, I think, the program tape. That was the main line of the program, and it was the only part that does not repeat, and it could call on any number of subroutines which were loop tapes. They had a different sort of physical implementation. You needed a different kind of tape reader for loop tapes, and the machine had one or two for the program tapes, and many more for subroutines. All these concepts had been quite clearly visualized. I also owe to Bell Laboratories a good way to draw a problem on paper. They came up with some worksheets, intuitively by experience, that listed instructions one after another, and along with each instruction on a line and next to it, there were perhaps twenty columns corresponding to the storage registered in the machine, so that with each instruction, you could write down what changes in storage it accomplished. So, at any time, you had a bird's-eye view of what was in store for the machine. You couldn't do that now with a thousand, or a hundred thousand words of storage, but it is possible to write down a problem like that, keeping track of the principal storage registers which are being changed by the routine all the time, and programmers now know that there are a very few of those. Most storage is in large batches, generic kind of things, where thousands of cells contain the same kind of information, and there are very few cells of what we might call scratch storage. It is a good way to present a problem by keeping track of scratch storage explicitly, along with the instructions themselves. Many little tricks like that, that we learned from the Bell Laboratories, were re-learned, painfully, later on. There really wasn't enough carry over, and that was because of a preoccupation with speed. That was justified because speed was the main feature of the electronic computers. But the people who handled them were preoccupied with speed to the exclusion of what they might have learned logically from the older, slower machines. I think that's all I have to say on that.

[End of Interview]