

An Interview with

DOUGLAS T. ROSS

OH 178

Conducted by Judy O'Neill

on

1 November 1989

Cambridge, MA

Charles Babbage Institute
The Center for the History of Information Processing
University of Minnesota, Minneapolis

Copyright, Charles Babbage Institute

Douglas T. Ross Interview
1 November 1989

Abstract

Ross begins with a discussion of his early involvement with John Ward on the Cape Cod project and his early experimentation in gestalt programming with Air Force and Emerson Electric Company programmers on the 1103 at Eglin Air Force Base. He then talks about the work he directed for the Air Force under Frank Reintjes at the MIT laboratories. He discusses the APT (Automatically Programmed Tools) and AED (Automated Engineering Design) projects which were early precursors of the languages and systems of modern CAD and CAM systems. Although these projects were not supported directly by DARPA, they were run in close connection with the WHIRLWIND, TX-0, TX-2, Project MAC, and CTSS projects that were running simultaneously at MIT. Ross discusses his use of programmers from industry in these projects. He also mentions the distribution of APT through the Fortran Monitor System and discusses the relationship between AED and PL-1. This interview was recorded as part of a research project on the influence of the Defense Advanced Research Projects Agency (DARPA) on the development of computer science in the United States.

DOUGLAS T. ROSS

DATE: 1 November 1989

INTERVIEWER: Judy O'Neill

LOCATION: Cambridge, MA

O'NEILL: I wanted to start out with a clarification question that I've been getting confused about, which is the Cape Cod system and where that fits in. That's pre-SAGE, right?

ROSS: Yes. Right.

O'NEILL: Was that part of your background? Did you work on that or you were just exposed to it?

ROSS: No. That was what I covered in the Personal Workstations paper. I and my group were the only ones, other than Lincoln Lab people, who used the big, complex display systems and so forth that they put in for the Cape Cod system. But we used it for our own personal workstation stuff on complex data reduction. That was the reason the equipment was there, for the Cape Cod, but we used it in a different way, a more general way.

O'NEILL: In the 25th Anniversary session at MIT you talked a little bit about Eglin Air Force Base in 1956, the 1103 that you had there, and the man/machine working together problems. Can you elaborate on that a little bit? What kind of problems were faced there?

ROSS: Again, my best treatment of this is in that Personal Workstation paper, where I went through the whole development sequence. But, I guess the main thing was - In what sense are you interested in it?

O'NEILL: I'm interested in the idea of the improvement to the man/machine interface, and the follow-up question to this has to do with the Licklider paper, "The Man/machine Symbiosis," and whether that was a summary of what was going on at the time as you saw it?

ROSS: Well, I don't know if I ever even read Lick's paper, actually. I presume it came after I had gotten started on this stuff because you see, by 1954, at least, well earlier--the timing is best dug out of that paper because my memory is terrible. The first paper I ever wrote was "Gestalt Programming" and that was in 1955. The whole idea there was to replace the laborious writing out of detailed programs and all those steps by having analyzed a problem area well enough so that you had what I later came to call a "systematized solution." Then you could compose different problems of this class by just plugging together pieces of program, and they would in turn be controlled by a push-button language. The user would make a number of discreet selections. It's just like nowadays it's done with *menus*, and when you had indicated all the pieces that you wanted to put together--by these mnemonic names and words for things associated with buttons, switches, with one meaning "period," essentially, for that sentence, you see--all these things would be brought together and that would be the man/machine, manual-intervention mode of problem-solving. I took over the term from studying Gestalt psychology, meaning that everything was brought together at once, as a unit, instead of this laborious step-by-step build-up. Even though I'd done the Scope Input Program before that, our resources were so slim that I hadn't looked around for particular things to do with it. In fact, we never did anything with it until much later. So the use of the scope was mainly for feedback display, but I *did* talk about a man/machine conversational mode of problem solving.

O'NEILL: Other people had mentioned the Licklider article as being quite influential.

ROSS: I don't think I ever read it, and I never got to know Lick until just before (i.e. in 1962 ?) we got into the time-sharing project (summer 1963), I believe. I knew a lot of other people, but I'd never run across him. He was at BBN at that point, I guess.

O'NEILL: Do you remember any formal influences to your concern about the man/machine interaction? Was it just your experience? Was anyone else talking it up or were you just trying to improve it for your own means?

ROSS: Well, it was that the problem we were solving was so intricate that we needed to be able to control what steps to take depending on what had previously happened, you see. The problem concerned evaluating the tail turret for

the B58 airplane, actually mounted on a B47 for tests. And we measured a lot of data in the air and put it on a magnetic tape and flew it down to play back into the ground computer.

O'NEILL: You transported the tape itself?

ROSS: Yes. Well, again, airborne telemetering at that point was too unreliable, and so we had this flyable Ampex tape recorder that didn't even start and stop! You just turned it on and off and it sort of coasted up and down. We made this whole box of equipment that would record all the things that needed to be measured forty times a second on this inch-wide magnetic tape and then you had a ground playback to be hooked up to the 1103. Then we would do the data reduction. These were extremely expensive test programs to run, with flight aircraft and all that sort of stuff--\$10,000 an hour is the number that I recall from it, which in those days was a lot. So the thing was that there were many dropouts and errors made, so we had to be able to adjust the very complicated data reduction programs that evaluated the test results depending upon what conditions we ran into. And so it was this need to have control over ongoing, vast complexity, rather than anything academic or theoretical that pushed me in those directions. Also, at that point, of course, I was very naive. It was my first job and I was the only mathematician in this hotbed of engineers and it was pretty far out. As soon as I saw the interactive display equipment, I said "Gee, that's just what we need."

O'NEILL: As a programmer, did you work on tools to make programming better for yourself? It sounds like your initial orientation was toward the end-user and getting people using it.

ROSS: At all stages, right from the very beginning, I tried to do things in a general way and then make it special for the use... That's just the way I've always gone about solving problems. I'm no good at puzzles. So instead of looking for the chink in the armor that will allow you to unlock something, I do best when I can envision a *generalization* of the problem, where things are sort of neat and abstract and idealized, and then get a solution there that I can then map down to a special case, you see? That's always been the way that I've done things. And also my training, which was still pretty raw, but it was all in pure mathematics, which is all about, you know, functions and systems and

making structures of things--abstract. And as I have also made the point in those history papers, in those days if you didn't have a very systematic way of doing something, you couldn't *do* anything because there were no resources. This was all done with 2K bytes of memory! (Laugh) Later it became 4K bytes.

O'NEILL: You were the Head of the Computer Applications Group of the ESL. What was the charter for that group? What were you trying to do?

ROSS: Well, actually, the group itself wasn't named or really formally became an entity until I inherited the APT numerical control programming project in the middle of 1956. This other data-reduction and tool-building project stuff that I had going was earlier. Again, I was the only mathematician in the Lab and my work got me into programming from the ground up. By then I had taken over the "computer" people, in the Lab. They ran Marchants and drew graphs and stuff. That's what I was hired to do in the first place, too. My wife did it for Lincoln Lab. So by then I had several staff members of my own, working on the B58 Project as well as doing other things. I did various other little projects that people would bring to me in the Lab. But by the time the APT Project came along, I was the only one that was a real candidate for leading the technical effort. So it was at that point that we gave me the title of Head of the Computer Applications Group; I chose that name. It ended up that for the next several years, through '58 or so, I had two or three sub-groups that were working on different projects but all interlocked together. By then we were not just doing the data reduction; we were building experimental programming systems --the SLURP System and all of its pieces; lots of tools for testing and writing and operating arbitrary programs.

O'NEILL: During this time did you have much interaction with people in industry? Were you looking at how people were using systems and interfacing computers with people out in industry?

ROSS: Well, only through these other projects. I mean, I used both Air Force and Emerson Electric Corporation programmers in doing the 1103 programming. Then we had contacts with the University of Texas on the ballistics theory, and that sort of thing and people down at the air base with respect to the operating of the computer and interfaced with Engineering Research Associates, which became the Univac Scientific group, when they got acquired

by Univac. I then also was active in various contacts with industry from committees. I was one of the founding members of the local Society for Industrial and Applied Mathematics (SIAM), for example. So, I knew people at Honeywell and places like that. And also I had been active with the ACM, and was the Publicity and Printing Chairman for two of the Joint Computer Conferences that were held in Boston at that time. So I knew quite a number of people around.

[INTERRUPTION]

O'NEILL: You were talking about your involvement with the larger community.

ROSS: Right. And then also, you see, although these were the particular projects that I was working on, there were probably a dozen other projects in the Laboratory, all of which I interacted with and the people. I mean, you know, control systems for an iron lung as well as the machine-tool stuff itself and all sorts of servo projects (the Lab was named The Servomechanisms Laboratory, originally) and this and that and the other thing. But my main feeling about things was this idea of making tools, making systematized solutions, and showing how everything we did was so wonderful it could be applied to lots of other things. That's what always drove me; that's why I called it the "applications" group. You see, we wanted to *apply* computers-- apply the tools and the methods to lots of other areas.

O'NEILL: Okay, so part of what you saw yourself doing was transferring of the technology or the ideas at least.

ROSS: Except nobody thought of these things technologically in those days. "Technology transfer" is quite a recent evolution in terms. It was just that you were doing your thing! I wonder whether historians can figure out when the idea of actually transferring technology to industry became a way that influenced the people who were doing it. For example, at the very heart of developing something like the machine tool was that industry would be the one to use it. But again the *reason* it was being done was because the Air Force had this very complex manufacturing problem and needed to promote better methods for doing it. Maybe I'm overstating it, but at least *my*

experience in the *MIT* environment was that people did it because the ideas and the *packaged understanding* is what they were doing--and so it's more of an entrepreneurial or a *publishing* thought, rather than a transfer-of-technology type of thing.

O'NEILL: So that wasn't an active goal.

ROSS: It may be that just this morning that it's affecting my thinking on it. (Laugh) I can't tell, because we certainly were very much aware that we needed to educate and do it and that it was called transferring technology. But I really don't think I would have used that term at all, or heard other people using it at the time.

O'NEILL: Were you familiar with what was going on in different places - for instance at General Motors? I know that they had had a computer aided design project that I haven't really looked into so I don't know a lot about it. But I was wondering if things like that or other projects that were going on in industry were generally known - were they part of this community?

ROSS: Yes, well, actually the General Motors computer aided design stuff came considerably after this period we've been talking about with the Whirlwind and early APT stuff. Again, through the computer conferences and later I became very active in the (ANSI/BEMA) standardization activity, and so from those we had contact with lots of places.

O'NEILL: Did you see a shift of this at all once you got involved in the ARPA funded projects? Was that more of an explicit thing in terms of getting out and spreading the word at the time?

ROSS: Well, not in my case because we'd been doing it well before the ARPA show started, you see. Also, my group never had any ARPA money. All I did was make deals to where I would supply my already-paid-for people and tools and programs if, out of the Project MAC funding side, would come office space and computer time and terminals. In fact, to this day, right up through SofTech, we've never really had *any* ARPA money flowing my way

directly.

O'NEILL: I was going to be getting into that because I was curious how your work was financed before ARPA, but it sounds like during that time as well - pre and during Project MAC - you were financed separately.

ROSS: Yes, because we had this direct Air Force funding all the time in various contracts and for various Air Force sources. And when we were using *Whirlwind*, Whirlwind time was also being sponsored and made available by the Office of Naval Research. We did use some of our own project money. When Lincoln took over, Lincoln supplied Whirlwind support to us free. But we did buy Computation Center time and buy from the IBM portion of the Computation Center, and at one point we also were one of the several sponsors of the Cooperative Computer Laboratory that had an IBM 709 that had its own sponsored projects. We'd gotten together [with the Physics Department] to do that and get more computer time.

O'NEILL: So you were always on the outside?

ROSS: With respect to ARPA, yes. Or a close colleague, I guess. We're sort of in the crack, you see. We never were in on the inside group for *any* of the computers we used. There always were in-house programming groups that we worked with very closely and they had higher privileges than we did at all times. But on the other hand, we were so useful and so pushy (laugh), at least I was, that we frequently would get priority treatments and be able to make deals.

O'NEILL: How many people are we talking about that were in your group?

ROSS: In my group? Oh, gosh, it ranged from the first round.... even the first year when I was hiring little part-time students to punch calculators, I had six or eight at that time. It usually ran from (early days) a half-a-dozen and then toward the end, 20 or 30 or more. I really don't know because we also had visiting staff from industry that were working on the APT stuff, the AED work later on, and some on the display stuff as well. It would get up to around 30

or so, and when you start talking about the Display Group--you see, later than the start of the Computer Applications Group, we also formed the Display Group which John Ward headed up, and that used some support from my projects' money and some from Project MAC. They were paying for both people and equipment in that (Display) case. So it gets sort of confusing.

O'NEILL: Were these people all programmers? I mean, were they all actually programming the machine?

ROSS: My people? Yes. [Actually, the larger numbers probably included some engineers, as well--doing adaptive control in APT days, and displays and light pens, in AED days.]

O'NEILL: So that was quite a large group.

ROSS: Yes. But, again we worked on several different things at once all the time.

O'NEILL: You were on the Long-Range Study Group. How did you get involved in that? Was it just because you were an active user?

ROSS: Yes. In fact I was trying to think.... Oh, one of the other things that I remembered to grab this morning was a copy of a thing I had done. This is with respect to just getting the graphics stuff out of my resumes. (I also have a fuller version that gives some display stuff.) It really isn't too relevant to what we're talking about here except as to what was going on. But I was trying to think where in my resumes I found references to the start of that committee. There are several times that I would talk with my boss, (Prof.) Frank Reintjes in the Electronic Systems Lab, about things that I thought we should get involved in with respect to computers that would crop up now and then and, surprisingly, ground work for *several* of those things came from my getting his approval to go out and rock boats and try to get something going. And I can't remember what the... it was probably from APT and my close ties with the Computation Center, because we were a big user there, that made it sort of natural for me to be included on that. But I really don't recall.

O'NEILL: Was your perspective coming in then, as a user, different from the rest of the people on the committee? Do you remember that being a factor?

ROSS: Well, we all were very different. (Laugh) Each of us brought a certain background of experience and things we thought were important and wanted to do and we were used to making deals and proposing things in rational ways so that those who held the purse strings and power of that sort would give us the go-ahead. That's sort of the way MIT runs anyhow. But (Prof.) Al Hill was the one who, I think, was behind the parent committee. Again, my memory is just too slushy. And then (Prof. Herb) Teager was the original chairman of it.

O'NEILL: I was wondering if the time sharing idea, which is obviously the recommendation of the committee, was something that you in particular were interested in. It sounds like some of your problems had to do with overall capacity and getting enough time to actually run your problems as opposed to time sharing.

ROSS: Well, again in the Workstations paper I cite that briefly. I didn't think it was going to be relevant, but maybe it is. Next week I go to Berlin to give an "early days of computer aided design" talk at a conference there, and when I was preparing for that, I ran across something.... I've put that in now I lost track of what I was going to refer to....

O'NEILL: We were talking about your computer needs versus time-sharing.

ROSS: Yes. Well you see, as the Workstations paper says and shows, the first keyboard input to a computer was when I had a Flexowriter hooked up to go with our data reduction programs in the secret room. And Arnie Siegel actually was the one who lashed that together for me. Although I think Corby remembers things slightly differently, and I've never checked with Herb Teager himself, there are several references in my resumes (and what I recall) that show the *starting* of the time-sharing idea took place in some meetings that we held with (John) McCarthy and (Marvin) Minsky and (Dean) Arden and me with people at Lincoln Lab, and.... Gee, I'm getting this all screwed up in sequence. But in any case, when Herb Teager was going to put the first Flexowriter onto the IBM 704, the resumes

show how we got ... I referred them to Arnie's earlier thing and they started off with that. There also is some reference there I ran across that John Ward may have had some interactions with Herb at that early handoff time. And maybe even Earl Pugh, who at that point was running the TX-0. He was in our Lab and when the TX-0 moved from Lincoln Lab down to campus, even though it was housed in Building 24, it was *our* responsibility and Earl Pugh ran that before Jack Dennis took over and then the PDP-1 as well. So in any case, the actual start up of the idea of time-sharing and how to get it going and so forth *does* have trails back into that very early stuff of mine on Whirlwind. I was up with it all along because, you see, we (John Ward and I) copied the design of the Whirlwind stuff to put a console on the 1103 in Florida, and that included a keyboard input. And then with Whirlwind on its way out and with the 704 having greater power, one of the multiple projects that my group did was to put the same light cannon and push buttons and indicator lights to hook on to the 704, 709, and so forth, and that's what later became the Display Group. But also, at the early stage, it overlapped with the Flexowriter keyboard input too. It was all sort of one package. [The actual sequence shows in the published Timeline of the MIT Computer-Aided Design Project.]

TAPE 1/SIDE 2

O'NEILL: The issue I was trying to get at was computer efficiency and capacity that were required by an application or kinds of applications you were doing versus the time-sharing idea of having the improved interaction with the programmer. Did you see a difference there or was time-sharing seen as just a step in the direction of getting better access - were there debates about efficiency and things like that?

ROSS: Oh sure, there was lots of that, but my own interest in it was, of course, that as long as you had the capacity in the machine to serve lots of people, then lots of people could do this Gestalt programming, experimental interactive program development, and use the displays and so forth. We planned from the beginning to carry over our display development and the light pen work and so forth. By then, when the TX-0 came down, we then had light pens to work with instead of the other stuff. So we were re-engineering the light pens, making a focusing one and improving the sensitivity and the electronics and so forth. We even made the innards for pens that we shipped out to Lincoln

Laboratory from our lab for the TX-2. So all the things that had to do with man/machine problem-solving was right along our line, and it seemed a very natural evolution to have the time-sharing idea come along. Again, that wasn't too much different from the communications work that was part of the Cape Cod/SAGE. So it seemed a very appropriate and natural thing to do.

O'NEILL: This kind of leads into another question I was going to bring up later, but this looks like a more appropriate place for it. There have been several comments along the lines of "Industry was slow in recognizing how important time-sharing was, and that it took a lot of effort and large lag time." Do you basically agree with that?

ROSS: Sure. Again, it's not merely that they were slow, but that they had a lot of investment in the batch processing--getting the maximum through-put out of expensive machines and a whole culture of that sort. I can't really recall when the shift came. It wasn't sharp at all, to where it was considered other than a pure expense, or maybe even a waste of money, to have computer resources used just for *programming*! See, it always was looked on as a burden. You were supposed to get through that stage and get to grinding those numbers out that people were paying for. So it wasn't clear that doing things like what we now call word-processing and document-preparation, and so forth, were what industry wanted to have their computers used for. And then to do more elaborate things, like the big engineering applications for aerospace and automotive and so forth--*there* you ran into the problems of inadequate display equipment and all that sort of thing. And again the culture of having to get things done on schedules, and they already knew how to do that from the batch processing. So it takes a while for things to get turned around. That's why I think some of the earliest places outside of research areas were in insurance and banking, because they *knew* they were paper shuffling, but they didn't have anything else, and it was their business, you see.

O'NEILL: Were the users or the programmers in industry--were there objections from them as well? Did they not see any use for it?

ROSS: I guess most of the people that I saw said, "That's great; when you've got all that government money at MIT,

do it." You see, they'd have to wait and see.

O'NEILL: So you really think there were financial constraints on their computer resources which made them focus on production instead of development tools?

ROSS: Right. And also the thing is that most manufacturing companies and so forth are actually very conservative because it takes a lot of investment to do something and they know the uphill battles. But on the other hand, if you looked into the various things going on in the SHARE committees and so forth, you can see that, especially from the aerospace side, and from the research end of the automotive, that the really big companies, right from the beginning were into the changes.

[INTERRUPTION]

O'NEILL: You mentioned SHARE in passing. Were you involved in that at all?

ROSS: Not directly. I attended various meetings, and later when they spun off a computer aided design series of conferences, I was at a lot of them. I don't think I ever had any SHARE membership, because SHARE membership came from the Comp Center. I wasn't active directly on committees.

O'NEILL: I'm trying to get a feel for the difference, if there was any, in the community at MIT versus the industrial community. We've already talked about that a little bit, they had to be more concerned with production runs. But in terms of the programmers who would be attending SHARE, for instance, was there a lot of interest in the interaction or were they just trying to get compilers to not have to run the machine language and things like that?

ROSS: Well, no, they had lots of things in SHARE. Corby would be a better one to talk to about what that is, or lots of different people. Their purpose, of course, was to influence what IBM would do to supply them better stuff. So they would have various levels of committees, some of which got to interact at a privileged level with IBM and

influence developments. But the main thrust, I think, was toward things that would increase the standard kinds of use, because that's where everybody had the most problems.

O'NEILL: It seemed to me that maybe SHARE would be some place where the time-sharing idea could've been pushed more within IBM early on, if that had been a priority.

ROSS: Well there actually was a competition with TSO and TSS, I think. Anyhow, there were a couple of other big time-sharing projects in IBM, and when they got started I'm not sure, because a big problem for IBM, as a result of our committee study and so forth, we chose to go with GE (or Honeywell) at that point. Heads really rolled at the vice-presidential level and so forth in IBM with *that* shock. But I was always on the periphery with respect to the other inside-IBM time-sharing projects.

O'NEILL: The committee that produced the report in 1961 about the need for a large machine, did that end there, or did you continue on with further activities in terms of actually getting into the MULTICS and looking at manufacturers?

ROSS: No. The committee wrapped up. In fact, I was surprised to discover that I was the actual editor of the final report from the committee, because by that time, we had essentially thrown Herb Teager out and rewritten the report. With the technical publications operation at ESL, I was the natural one to draw it all together and get the report printed. So, as I recall, there was nothing formal that happened after that. Also, I don't remember what kind of advisory committee structure or whatever Bob Fano may have had at the beginning of Project MAC and whether John Ward or Frank Reintjes were on that sort of thing. I don't think I was.

O'NEILL: So your direct involvement ended with the report.

ROSS: Yes. I think so, but I can't really be positive about that. I remember going around and visiting on a confidential basis. Members of the committee would go around and meet with the major manufacturers, Burroughs as

well as IBM and GE.

O'NEILL: So you did that before the report was finished?

ROSS: That's what I can't remember. I think that was all done.... I'm pretty sure it was pre-Project MAC.

O'NEILL: The report itself says you haven't talked to a lot of manufacturers.

ROSS: There may have been this follow-up function, and until I find it in my trail of papers (laugh) I might as well be somebody else. Have you talked to Corbato yet?

O'NEILL: I personally have not, although Arthur Norberg has interviewed him and is going back for a second interview, so I'm sure that's one of the areas that will get covered. You talked about looking at other manufacturers. Do you recall looking at other time-sharing systems at the time--were you familiar with the Dartmouth System?

ROSS: There weren't any.

O'NEILL: I guess that would be a little later wouldn't it?

ROSS: Yes. Right. I didn't know anything about the Dartmouth effort.

O'NEILL: Or things about the SDC system? Or was that later?

ROSS: SDC with JOSS and so forth I did know more about. I knew the people that were doing it. I listened to Jules Schwartz... I can't remember right now, but I'm not sure through what channels I had this, whether it was from the Standards work, or what. Of course, they were included in the summer of 1963 kickoff of Project MAC.

O'NEILL: What was your involvement in that summer session?

ROSS: I was here. (Laugh) And active with Dr. Wilkes, Strachey, and people like that when they were here. But I don't recall a great deal about it at this point.

O'NEILL: I didn't recall seeing your name on any of the reports, that's why I asked.

ROSS: Again, I'm not sure myself when I moved my group over to these offices, right here on the fifth floor. It may be that that move didn't take place until the summer session was finished, in which case, I would've had my own office base in Building 32. I would come over for various MAC activities and meetings, but also, I had my project to do so it wasn't the same as being one of the full-time visitors who were involved in that (writing MAC papers).

O'NEILL: You mentioned that you didn't have any direct ARPA funding, so you weren't involved in writing any proposals or any of that.

ROSS: I do remember when Licklider first came and tried to get us to take a Q-32 off his hands.

O'NEILL: Oh. The machine SDC had.

ROSS: Right, that's it. They had a leftover one of those, and again there's some reference to that particular meeting in my resumes.

O'NEILL: It sounds like there wasn't a lot of enthusiasm for that particular idea.

ROSS: I remember talking with Lick not too long ago about that a bit. And, he couldn't recall either, precisely whether it was a straw man to get us to react or what, but it did happen. (Laugh)

O'NEILL: Sounds like he did get a reaction. What I was leading up to before was we talked a little bit about how your work was funded with the Air Force. Did you actually write formal proposals to them?

ROSS: Yes. The Laboratory did. You see, I never liked administration stuff, and so, for example, even on the APT Project, when it started off, John Ward had been the Project Engineer on the B58 Project when I started, so he was my boss all along. Then he moved up to be Associate Director, at one point he was Executive Officer, and so forth, for the Lab-as-a-whole, in addition having these project responsibilities, you see. So when the APT Project needed to start, he was Acting Project Engineer for that project. Then we moved Don Clements, who is an engineer's engineer, not a programmer of any sort, to be Project Engineer for the APT Project. He was until... I forget exactly. He had some way-out, flaky things he wanted to do, (laugh) and I actually *did* everything [leading the technical work], so he had a sort of hard spot to be in--to be a Project Engineer on a job where I was actually running everything except for the paper work he was doing, and what he was trying to do. So he sort of got frustrated and he phased out. But when the Computer-Aided Design Project came along, (the same sponsors but just a new contract, a new mission, you see) we were going to sub-contract from that to the Mechanical Engineering Department to get some actual designers involved. We needed to have a Project Engineer. Frank said I had to be Project Engineer. And I said, "Well, I still have all this B58 stuff running around, we've got our SLURP system for our automatic program development, we've got all these other projects, we've got light pen development, we're just getting phased into Bomber Defense," (another phase of John Ward's thing)--again, I had about five or six projects going on, some of them rather hot and big. Plus, all my *other* running around. So I said, "Gee, I'll be Project Engineer if Dick Osborne (who was the Executive Officer for the Lab) will do all the paper work." So that was my deal until I left MIT. Even though I was Project Engineer, I never did any of the budgeting or whatever. I just kept doing my same old thing. Which wasn't really the best background for starting a company. (Laugh)

O'NEILL: Although you didn't have administrative responsibilities, did you have a lot of interface with the Air Force?

ROSS: No. As a matter of fact, the way that worked was we had for the APT and computer-aided design stuff (which

now was the sole focus other than my interaction with Fano and his sponsors)--that had been on continual stream of Air Force sponsorship from 1950 or '49 on. From the very early days, Bill Webster was the Air Force engineer, a civilian, who monitored and granted the money. So we had this very closed working relationship, because everything had always been so well accepted in industry, both the machine tool itself, and then APT, and then the computer-aided design stuff itself. As soon as I came out of the closet with it [we purposely went quiet, after the world-wide APT publicity], we started the AED Cooperative Program right away, where industry would actually send, at their expense, people to work for my project. They didn't bring problems in; they just brought their warm bodies.

O'NEILL: What was the name of that project again?

ROSS: This is AED, Automated Engineering Design. [I didn't like the sound of "CAD," originally--and "AED" = "aid" sounded *helpful!*]

O'NEILL: Oh. AED, okay.

ROSS: The Computer-Aided Design Project's main part was that. Again, it was just deals that I made. See for years, I had said that I won't abuse students and exploit them as slave labor, which is very unusual around MIT. (Laugh) I also wouldn't hire people who were not fully qualified, who really couldn't *do* the job. So for years we'd make these proposals, get money from the Air Force to do things, and I would then request a "no-cost extension," because I hadn't been able to spend the money yet. It was quite different. Most people would spend it and ask for more! (Laugh) But because they understood what I was doing, both the MIT and the Air Force people were quite pleased to operate that way. But it meant that we had many more things that we knew *how* to do, research and development that *could* be done, than we had *ability* to do because of lack of [qualified] people. So right at the very start of the APT project, I had gotten Frank Reintjes' approval to invite... When [early spring, 1957] I went back to my second meeting with the Aircraft Industries Association's numerical-control subcommittee, because I knew what pressures they were under to get things going, I went to that very second meeting with a proposal to them, that if they could send some people, (I couldn't find them or hire them) if they could send some of their experienced people to work with

my people at MIT for a while, using Whirlwind and so forth (the Computation Center 704 was getting started, you see), that I was sure we could make an APT System for them to get them off the ground. Their reply was that they were in a big economic crunch, counting pencils and so forth. So their reply was no, we can't afford that, but can't you cut the project up in pieces, and we'll work at the pieces at home, send them to you at MIT and you can put them together, tune it up, and ship it back. So that's how the whole APT thing went, you see. So all the time, I was using outside staff, deeply experienced people, to augment that I was able to put together at MIT. That came in *spades* once we got the AED Cooperative Program going, because that ran for a whole five-year period. It had people all the way from half a year to 14 or 18 months at a time, just moved lock, stock, and barrel. North American moved a guy, his wife, and eight kids, and his mother from California here for 11 months! (Laugh)

O'NEILL: That's amazing.

ROSS: So anyhow, that's how I solved that manpower problem.

O'NEILL: Did they have the facilities available at these various locations?

ROSS: No. But they all had 701s, 704s to start with, and they were SHARE members, you see and so forth.

O'NEILL: So they did the work on those. Okay.

ROSS: Again, we use both our contacts with the Computation Center as well as directly to the local IBM office to solve our problems with respect to doing things with industry, with respect to IBM machines. For example, when we did our first distributions of the APT System, we chose to do it through the Fortran Monitor System (FMS), which had come out of SHARE and was IBM supported. And of course, it turned out that nothing as big as APT had ever tried to go through that. So, we had to get in and re-work all the tables and become experts at all sorts of things.

O'NEILL: Yes, it's amazing what happens when you stress a piece of software.

ROSS: Yes, we did that all the time.

O'NEILL: Who paid for that?

ROSS: Who paid for what? The IBM work for example? Again, I would just make deals.

O'NEILL: You didn't have to worry about that?

ROSS: I did, you see. That's right, this whole chain of discussion came from your asking how we got the money. Well, with respect to the Air Force money, all I did was make the things come out right and make Bill Webster feel good about it, because the industry felt good about it, and when it came time to write a new contract, I'd sit down with John Ward and Frank Reintjes, and Dick Osborne, or whatever, and we'd work out a one- or two-page, just bullet items, of what should be done next. And they would send that off to Wright Field. Frank would go out and visit with the colonels and so forth, out there once in a while. I never had to, never did. So pretty soon this one- or two-pages would come back as our contract.

O'NEILL: So it wasn't really blown into a full-fledged proposal?

ROSS: Not by today's standards. We didn't have to. It was a very, very close relationship by then, because they counted on me at the technical level to both lead and respond to the industry people as we needed it. That's all the Air Force wanted out of it. So they were happy to contract for the whole thing. And then when it would come to things that weren't in our budget, which was all... I mean we did put in amounts for buying equipment to do the light pens and that sort of thing. Or we did development of adaptive control for the machine tool and hardware things of that sort, those were also going on. But when it came to a big crunch about how to support the Industry APT Project, I would just go and talk with Loren Bullock who was running the IBM show. I'd say, "Hey, here's what we need, Loren." They had a whole third of the machine at the Comp Center, for IBM use, you see. So I didn't even

have to point out to him, you know, that these were all very important IBM customers. It's just that we had these very good working relations. So they'd make the IBM time available to us when they could, at no charge. I don't even remember who paid for shipping all the boxes, by railway express-- boxes and boxes of cards! That was how it went out. A whole *pallet-load* of boxes of cards, to 19 places or something. (Laugh) I don't remember who paid for that, whether it came out of our project or what.

O'NEILL: But industry was at least providing the people and the facilities.

ROSS: Yes, right. Also when they visited. APT was done with assignments to individual companies, but we had to meet frequently in different places, and they'd pay all their own expenses. I and my small staff--we only had, other than Clare [Clarence G.] Feldmann... Well, Jerry Wenker, I guess... It's sort of hard... I never had more than two or three of my *own* programmers working on APT exclusively--the *industry* part. Except Clare, who was my one prime person. So the same deal-making style, you see, is what I carried on once Fano got set up with his funding for Project MAC. The AED Cooperative Program itself, again, the same thing was happening in spades. The same scheme that we knew, only much more. I could lead much more technological development than I could possibly do with MIT staff. I couldn't use students: It was too difficult, required too much experience, and was too down-to-earth, sort of, to get good thesis material out of for them. So I couldn't use graduate students properly, and I couldn't hire the staff because we paid less. It was so wonderful to be MIT, you didn't have to pay as much, you see.

(Laugh) I can remember having a conference call on the telephone from the conference room on the 8th floor here in Tech Square, in which it went out to a whole string of big aerospace companies. It wasn't for starting up, but it was for doing the next round, I guess--the following round of the AED Cooperative Program. By the time the conference call was through, which was essentially my brow-beating them into it, nobody was willing to say, well, they were going to be the cheap one. (Laugh) Every one of them was saying "Yes, we'll have somebody there." So I essentially snookered this whole bunch of people out of at least \$50,000 a pop from each one of them, completely informally. I mean, nowadays, you go through all sorts of commotion to set up such things. But in those days, I just said "Well, I've got the office space," in fact, you can see this size office, it's cozy for *two*. I had *four* in them, because I refused to split up into different floors, even if the space were made available to me! I liked to have my

people working side by side; they *worked* better. (Laugh)

O'NEILL: Was that something you learned through trial and error? I mean, had you dispersed projects before?

ROSS: No, I just knew from the way the work was done. It was done by know-how seniority. In other words, whoever knew more automatically would gravitate to be the one that would lead different portions of things. What continued to astound people, and me, was that my own MIT people who had been with me for a long time, inevitably *were* those leaders, even when you take people with more responsibilities and supposedly bigger people from industry. They'd come in here and just couldn't keep up with our normal way of doing things. They learned like crazy from us, and grew. Then, when things got more refined, then I would be able to give certain of them the lead for some tasks.

TAPE 2/SIDE 1

O'NEILL: We're talking about the industry people coming in and becoming leaders towards the end of their time here.

ROSS: Yes. There were various portions of the AED thing where I was able to not have an MIT person leading that task. For example, the first use of *finite-state-machine* methods for lexical processing, which is now the standard way of doing all language processing--we had the first one of those for AED. How I got that done was I worked with three industry people; there was an ACM paper on it. They were able to work together well enough, and the little amount of time that I could spend to actually work with them, and lay it out, and this and that, and the other thing, answer questions and prod--we made a very successful piece of it.

O'NEILL: So you viewed this as a training procedure for these people?

ROSS: Oh, yes. Oh, definitely. When they came in we organized all-week workshops. I mean, they had lectures in

the morning from me, then project assignments in the afternoon, and it was a real hotbed. Again, they brought no problems of their own here, to just *use* the environment, which is what a lot of other MAC visitors did. These were my *employees*, as far as getting stuff done. That's not really bad, you see. I always did things by what I called "semi-democratic" methods. I first used that at APT Kickoff Meeting in May 1957, where I said "We'll run this thing by semi-democratic authority. That means we have full open discussion and democracy as long as I think things are going right, and when I see a problem, we're going to do it *my way!*" (Laugh) So everything always ran that way, you see. So they just became integrated, became a part of the group.

O'NEILL: Did you have any control over who came?

ROSS: Only by specifying the arduous requirements that they would meet. Some places went through a very formal process, but they all sent the very best people. I think it was Dow Chemical that had 20 candidates, and they ranked them all under 12 or 16 categories, and there was one guy who was tops in every category except one, and he's the one they sent. The category he was *not* tops in was *availability*. (Laugh) So, I mean, it was quite a prestigious thing to be selected by your company to do this. And, again, you see, many of the companies had previously gone through two or three rounds of similar, what the British call "seconding" of staff through the APT years. Because APT, even after we got out of it and turned it over to industry, it kept going, and they did finally get everybody together in one place in San Diego. Clare Feldmann used to go out there for months at a time, on the-wrap up of the first phases of APT. And they kept going with this group effort, with various industry leaders for quite a number of years. So it was that background that made the AED effort follow on very naturally.

O'NEILL: One of the objectives listed for Project MAC was the training and education of people. It sounds like you had a different approach to that in terms of training industry.

ROSS: Again, that's the idea of the Computer Applications Group, which is still the umbrella under which all this was done. Our mission, my thought about the whole thing, was that we wouldn't do way-out, theoretical research. I used to not *want* to do artificial intelligence, for example, because the artificial intelligence people, the artificial intelligencia,

were all my very good friends, and so forth, but they kept choosing little games to play, and little things that they could master, right? But my whole philosophy has always been give me a really tough problem that's just beyond the state-of-the-art, and give me a whole bunch of users beating on me to get it done. In other words, the real core of what being an *engineer* is. You have a scientific basis, but when you don't have the science, you put in some bugger factors, some safety factors, and so forth and you get smart enough, and you get the job done anyhow, right? Economically, and as close to on time as you can make it. And if the customer is asking for something that's outlandish, give him what you can, and educate him back to what it is. Or if you thought you could do so much, and when you get in there and look at it, you can't do that--*down-sell*, and build up what you *can* deliver. That's the way we did the education. We *never* were able to complete what we *said* we were going to do, because all the time, as you get into *doing* it, there were other things that were more fundamental and more worthwhile, and so it would build a better base, and you wouldn't get to where you *thought* you were going, but in a little while, you were *further* than that! We reworked each component of AED at least a *dozen* times! Things were done very modular, with building blocks-- what we called "integrated packages" in the AED context --and with sub-systems and so forth. The system was all built by itself, with itself. So we were our own heaviest user, so you get lots of bugs shaken down. We didn't put many bugs in, but you'd find that it had new behavioral characteristics that you hadn't foreseen, until you refine it, refine it, refine it. Most of the major library components in the AED System have been reworked at least a dozen times.

O'NEILL: And were constantly in use while you were building?

ROSS: Yes. In fact, I calibrated it at one point, and the system had a half-life of about three and a half months. In other words, every three and a half months, 50% of what was then running was brand new code, as the system expanded as well. So it's in that way that we had an evolution toward extremely smooth and reliable operation of all sorts very exotic stuff. *Still* some of the things we did in AED haven't been done since [in current software-engineering systems].

O'NEILL: Yes. I know you were instrumental in putting up some of the early software engineering conferences.

ROSS: Well, [in spring 1968] I taught the first software engineering course [in the *world*] here, at MIT--to get my book written about how to use AED. I came out with this book, *Software Engineering with the AED-0 Language*, which tried to package that up. Again, you see, that was after we'd been heavily developing and using the stuff for five years.

O'NEILL: Okay. So you had a five-year shake-down period.

ROSS: You see, we not only had all the industry people, but a very large number, we never added it up (I'll have to do that sometime), of MIT project users that built all of their systems under AED. Forrester's Dynamo feedback-modeling systems, and the econometric modeling systems that the National Bureau of Econometric Research still runs [the Troll System], both mechanical and electronic computer-aided design stuff...

O'NEILL: So this system was being used?

ROSS: Oh, all over. Because, you see, one of the first things we did was make sure we had compatibility with MAC standards. I backed out of the theoretically-beautiful and clean way that we originally were generating AED code and had the whole compiler back-end redone. We did what we called a *disassembly*. We took the machine code of the compiler, itself, (just a bit- map of memory essentially) and went back as though it had been written in the FAP assembly language, then came back out and reassembled the whole thing so that it would be easier to do patches. Then we changed the whole code- generation part of the compiler to generate the standard (what was called the *BBS format* of machine-language representation, so that whatever came out of our compiler would fit into the standards that they'd adopted for Project MAC.

O'NEILL: Were the people working on this project using the Project MAC computer exclusively, or were you using whatever you could find?

ROSS: No. At this stage, we were all using the MAC stuff here. Well, when we got to doing the distributions, no, not so. Later we got to using everything all over the place. You see, we did both timesharing and batch processing versions of large AED releases, and also from the beginning, AED was a machine independent software engineering job. In fact, we started out in parallel on the 709 and the TX-0, before MAC was even thought of, you see. That was actually the Bootstrap Compiler--in which we first wrote AED [we didn't use FAP]. Then it was that first AED Compiler that we had up and running here within two weeks of Project MAC opening its doors. So when we were doing these distributions, or later, we got to doing what we called "half-bootstraps," cross-compilers nowadays. In particular when we went from the IBM 7090 to the IBM 360, which is a big jolt, going from 36 bit words to 32, and all sorts of things, you see. Then, for various stages of that, we would go around and use computers all over the place because different ones would have different parts of the macro processing system, or somebody would go from this kind of tape to that kind of tape without this and.... it was just unbelievable. But we did it. We also had a half-bootstrap going to MULTICS as well, which never got completed all the way. We did a PDP-10 at the same time. The first working *complete* bootstrap was to the UNIVAC 1108, even though the only machine that was available was down at United Aircraft. [Bob Coe led that effort when he went back and did major chunks of the support of that effort here at MIT as well--in parallel with our 360 Bootstrap.]

O'NEILL: So from the beginning it was designed to run on multiple machines and it was successfully run.

ROSS: Only very recently is there any approaching the portability that we had. The C Language is considered a portable system, and UNIX is "portable," but neither matches AED portability. Well, UNIX actually did complete many of the things that we wanted to do with respect to the total system. We never had what was the MAC side of it; MAC never had control of it (laugh). So UNIX ended up putting in many of the things that I wanted to have. In fact, some of the... I don't know whether Corby would agree with this, but some of the main features of the file system of MULTICS that got carried over into UNIX and so forth, and a number of protection things were almost a direct offshoot of the dynamic free-storage handling of the AED System, and that's the way it always seemed to me--as just an extension of the storage system. So in any case, there are cultural ties there too, I guess.

O'NEILL: It strikes me that perhaps had better networking capabilities been available sooner, your job would have been a lot easier.

ROSS: On these machine transfers? Oh, yes. Well, again, many of the things *still* haven't been properly solved, I understand. You still have all sorts of protocol and data-formats problems. It still is not well worked out.

O'NEILL: Did you use the ARPANET when it became available?

ROSS: No. ARPANET came after we were either on our way out, or were too wrapped up with other things.

O'NEILL: So it was too late to have any affect on you?

ROSS: Yes, right. Again, we left in summer '69. I don't remember when ARPANET started up. I think it was starting up around that period.

O'NEILL: It seemed like, especially with people all over the country working on the same project, that ARPANET would have been useful.

ROSS: Well, it was a real disappointment to me that the MULTICS development was done in PL-1 rather than AED. I offered several times to Corby to merge our group efforts and make all our knowhow about building systems with a higher level language and all that sort of stuff, software engineering (we didn't have that name for it then). But he felt very strongly the responsibility he had to deliver, and so he said, "Nobody's going to work on the innards of this project unless they work for me." And I said, "Well Corby, I don't work for you, I'm not going to, but I still would like to have us collaborate. I'm on the opposite side of the same floor of the building with you. (Laugh) You know where to find me." But then they got in bed with the Bell Labs people and went all gaga over PL-1, and even though I knew more about PL-1 and its problems [from my Standards Committee work]... In fact, one of these papers... I should send a... I don't think I've even done that. There's a series of historical papers I've done this year by invitation,

which I should send out to Babbage, just so you have it to see what I've been doing. One of those... what was I talking about... I shouldn't take time out to fill in those transitions. (Laugh)

O'NEILL: The PL-1 issue.

ROSS: Oh, yes. One of those papers of mine *does* cite an IBM study that Jean Sammet had done by Bob Tabory in her group. Nat Rochester's group was here on the third floor, I think, and Norm Rasmussen's group was on the fourth floor, which is where we are now. *We* were on the fifth, and so forth. So Jean worked for Nat. In her group she had Bob Tabory do a study, a comparison, whatever, of AED and its effect on PL-1. So there is this IBM report that covers that, somewhat. I mean there are things in it that I didn't agree altogether with. But in any case, *IBM* was not all hung up about the relationship between PL-1 and AED. But everybody else seemed to be. So we just went off and did our thing, and they went off and learned a lot the hard way that we could have told them. (Laugh)

O'NEILL: So you weren't involved formally at all. I mean, they were just doing their own thing separately, and informally you had some connection, but nothing formal.

ROSS: That's right. I mean, there was a big cultural difference. For example, when Project MAC was new, there was a series of seminars, or afternoon things, and Corby asked me would I come and tell them about what the AED Project was doing. So I said, great, sure. The first seminar, lecture, talk, I started out by describing the overall goals, and then jumped right in and said the most important things is the *Symbol Table*--and spent the rest of the time talking about the symbol tables, and I got completely lynched! Nobody could see... *They* wanted to hear about everything *else*. What's a symbol table? They didn't see any reason for that. Well, it's the core of it all, at least what... Nowadays, *everybody* looks and says databases are important and all that. In other words, it had the same role for what compiling was about, but nobody had *done* that before, and, somehow, they couldn't bridge this gap and got impatient. So after a little while, that attempt to cross-fertilize sort of poohed out because we were just on different wavelengths. Later on when they were building MULTICS, they had great problems. It was a huge project. Very, very nicely developed. All sorts of methodology and stuff that worked. But when I'd hear (I never got enough

details to know, but...) when I'd hear about things I'd say, "Oh well, we could've warned them about that." At one point somebody, it was Corby or one of his people, maybe it was Bob Graham or Bob Daley, was saying to me about it, "Gee, we finally got this key module reprogrammed and it's great! We're really making progress now. It works a *hundred times faster*." I said, "Gee, if *I* had anybody who could rewrite *their* program and it would run a hundred times faster, I'd fire the bastard." (Laugh) So there's the difference.

O'NEILL: Did you personally use the MULTICS system, or just people in your group?

ROSS: No. In fact the group never did very much.

O'NEILL: Oh, I thought you'd said before...

ROSS: Well, we started a half bootstrap, a cross compiler for it. But we never used it, because we moved out too soon. Again, our last activities were packaging up the 7090 and 360 systems for distribution. I don't remember when MULTICS came up and was available, but I don't think we actually had anything active on it.

O'NEILL: You did use CTSS, though?

ROSS: All the time.

O'NEILL: Were you involved in writing, or the people in your group, making extensions to CTSS as users?

ROSS: All the time. From the beginning. In fact, many of the tools that were used, incorporated in the system files, were done by AED and its users.

O'NEILL: I had seen some early references, did that continue on?

ROSS: Yes. I never made a study of which, you know how much, and all that sort of stuff. They also had MAD, it was another language that was used. Bob Graham came from Michigan, I believe, so there was a strong interest in MAD.

O'NEILL: Yes, I've seen it mentioned as being one of the languages available. We've pretty much covered the set of questions I had. As you know, the study we're doing is with DARPA, so this has given me an alternate way of looking at things by speaking with someone who wasn't completely involved in the DARPA part of it. Is there anything you'd like to add to the record?

ROSS: I do much better in a response mode, even though my memory fails me. No, I guess I would just say that so many of the things that I've been exposed to or part of around MIT have this flavor of having really excellent people given lots of freedom, rope to go hang themselves, overreach, but work very, very hard and get outstanding results. Again, maybe a little different than you thought it would be, but never a real flop. And never a waste of time. Even though relatively large amounts of money were involved, for the times, so were the results, so were the expectations. So I see that whole way of doing things as being of the very essence of the fabric of MIT. I think it was really brought into focus in the success of the Rad Lab effort. Before that, MIT itself was pretty green. The fact that that was pulled off and laid such solid groundwork is what has carried through. And, although I never talked to Lick (I see more of Lick now than I ever used to, just for chats and things occasionally), but I never knew him in the early stages, you see. I ran into other people, but never ran across him.

TAPE 2/SIDE 2

ROSS: So, in any case, whatever Lick's actual background, I know he's a psychology/behavioral science type. He came through BBN. I don't know where he was before that, for example.

O'NEILL: He spent some time at Lincoln Labs, too.

ROSS: Was that before BBN?

O'NEILL: Yes.

ROSS: So where he was before Lincoln, I don't know. Because, you see, Lincoln was just coming on when I started. But in any case, I know that Lick and the people who followed him, Sutherland, I guess, was the very next, and then Bob Taylor, Larry Roberts. I can't remember whether Bob Taylor ever had a direct MIT link, certainly Roberts did. So this sort of way of doing things, which was really being a part of MIT, which was being a part of *the* world, (Laugh) where the really hot stuff, the technologically great stuff comes from. It still permeates the whole place. You can't think straight because of it. It really warps you. But it's that that epitomizes the whole thrust of what you're studying. I mean, the ARPA sponsorship, right up to today, is that same family, you see. And again, it goes back probably to the ONR roots. Again, I've read just little pieces about that, but the Vannevar Bush start, and whenever ONR got its style, those are what make it go.

O'NEILL: So you just see that going on, progressing?

ROSS: Yes. I think in later years, where people started to try to turn to ARPA for actual specific focus on problem solving... There was one stage where everything had to be associated to some particular weapon system. Those distortions, you see, did distort at various stages, and have left their mark, I think... I don't think the DARPA culture is as pure as it used to be. It *can't* be with the budget crunches and all of the mismanagement that comes in from other places, and high pressures. I think it's a part of this very important, very American, if you will, combination of government and academe and industry working together to get things done. I guess the thing that I like about my non-DARPA ties, (it's part of the same picture, you see)... mine is sort of more pure in the actual bringing in of industry. The actual treating the end customer as the *reason* for it all--that it's essential that they be involved all along, and that they be educated and upgraded thereby. You see, it's not just a transfer of technology as dishing up some beans onto a homeless person's plate. (Laugh) Which is what too many other things come from, or it turns around the other way and becomes a big grab bag where the industry side gets too organized and too greedy and too

shady, and starts picking the pockets of the public, which also goes on. So there are all sorts of ugly distortions that have come in the last decade or so to what, otherwise, I think is a very significant measure of what it has been to be a technological part of America. It's quite characteristic, not duplicated any place else.

O'NEILL: I wonder if, when you were doing the cross-industry work, bringing in industry, was there ever any problem between the participants in terms of not wanting to share anything they were doing?

ROSS: No. Because, again, when they came here they were just here so absorbed in what we were doing, and worked so hard, that many of them said they'd never been in such a hotbed before. They all went on and went to positions of considerable responsibility, and so forth, because of it. That also is, I think, what keeps Lincoln Lab, MITRE, AI Lab, and LCS here--the current pieces that continue from this whole thrust--they are national assets, each one of them. Again, I think in the old days it was a lot cleaner, but it's been a good thing to be a part of.

O'NEILL: Okay. Thank you.

END OF INTERVIEW