Bay Area Computer History Perspectives
and
The Computer Museum History Center
present:

# "VIGILANCE AND VACUUM TUBES: THE SAGE SYSTEM, 1956-63"

5:30 - 7:00 PM
Tuesday, May 19, 1998
The Computer Museum History Center
Building 126
Moffett Field
Mountain View, CA 94035

In 1963, the last of 22 SAGE command centers was completed by contractors IBM, Western Electric, The RAND Corporation, and Burroughs. At a cost of $8 billion (1964 dollars), this vastly complex technological system, an outgrowth of MIT Lincoln Labs' Whirlwind II computer, represented the state of the art in strategic doctrine and computer systems design. Each one of the 22 SAGE command centers used over 49,000 vacuum tubes, weighed 250 tons, and consumed 3,000,000 watts of power.

The SAGE system linked these command centers into a technopolitical "shield" against Soviet strategic bomber attack.  From a stark social context of high Cold War tensions emerged impressive technical advances in hardware and software systems design, real-time control, and air traffic monitoring.

Advances such as the light gun, modems, duplex CPUs, multiprocessing, A/D and D/A conversion techniques, as well as networking arose as ancillary technologies of SAGE development.  But did SAGE really work as advertised? Should we care?  This lecture reflects on these questions, SAGE's context, and its technical spinoffs.

The lecture takes place in front of 400 square feet of actual SAGE hardware, including Weapons Director and Intercept Technician consoles! This equipment is from the last functioning SAGE center in North Bay, Ontario (Canada), decommissioned in 1982. The USAF SAGE Film "In Your Defense" will also be shown.  "I like Ike" buttons optional.

For more information, including photos and audio soundtracks of the SAGE system,  please visit our new Computer History Lecture Series web site at:

http://www.computerhistory.org/sage

The Speakers:

This lecture's speakers represent a variety of perspectives, from the history of technology, to hardware and software systems engineering:

Les Earnest: Senior Research Scientist Emeritus, Stanford University, Project Engineer and System Designer, SAGE system hardware.  Founding President, Imagen Corporation; former Associate Chairman, Stanford University Computer Science Department; Executive Officer, Stanford AI Lab; Department Head, Information Systems Dept, MITRE Corporation; Member, Technical Staff, MIT Lincoln Laboratory... and inventor of the original (DEC-10/20) FINGER program!

James Wong:  Computer Systems Engineer, Burroughs Corporation; Unisys Corporation; Project Engineer on SAGE system software for The RAND Corporation 1955-1963; Team Leader, System Development Corporation (SDC), Lincoln Laboratory, SAGE and Project "465-L." Mathematician and programmer for the IBM CPC, 701, and RAND JOHNNIAC. Wong is retired and currently volunteers as an instructor in Mathematics with the Learning Disabled Program at Foothill College.

Paul Edwards: Senior Research Scholar and Lecturer, Program in Science, Technology & Society, Stanford University; author of "The Closed World: Computers and the Politics of Discourse in Cold War America."  Edwards has also authored dozens of articles on the history of computing and has held visiting professorships at Stanford, Cornell, the University of Michigan and UC - Santa Cruz. His next book is entitled: "The World in a Machine: Computer Models, Data Networks, and Global Atmospheric Politics." Edwards will be making a 30-minute presentation.

These talks are sponsored by
The Computer Museum History Center
and
Sun Microsystems

PETER NURKSE:

Hello, I’m Peter Nurkse from Sun Microsystems, and some of you I’ve seen coming to these talks for five years! You know, it’s been five years now that we’ve been holding these Bay Area Computer History Perspectives talks, and with the Computer Museum now since the arrival of the Computer Museum from Boston. I remember right now -- what I’m thinking of is our last talk at the end of March, which was so cold, it was freezing! That was, before the end of March would be a reasonable time to have a talk here in an unheated area, but this year hasn't been an unusual year. And then, thinking farther back, a talk that comes to mind is ILLIAC IV, we had a talk on ILLIAC IV here at NASA, and I remember in particular how we were all amazed that ILLIAC IV consumed 250,000 watts, just ILLIAC IV itself, and another 250,000 watts to try and cool it.

So now, here we have SAGE which consumed one million watts, that puts ILLIAC IV in its place. One million watts, that was just for one computer, and there were two computers in each SAGE center, and there were 22 SAGE centers around the country. So you can figure how many megawatts were going into SAGE, just out of the domestic electrical production.

And this is really special, because we have the speakers talking right in front of pieces of SAGE, which is a huge computer -- there's only small pieces of it here. And when we project the video, the screen isn't quite big enough so parts of the video will be projected onto SAGE itself! [Laughter] SAGE is really here this evening!

So we'll start with the video, which is the classic Air Force film of the era, and then Paul Edwards will speak. As I checked before we began, Paul was age 1 in 1958, when SAGE was first implemented! But, Paul has observations on SAGE. And this is example history, each generation comes back and looks at history, and in computer history we're already at the stage where we have a second or a third generation coming back and looking at events and what happened.

And then we have two veterans of the era: Les Earnest, who'll be talking on hardware, and Jim Wong, who'll be talking on software. So you'll really have the overall picture of SAGE.

I'll mention, our next program is scheduled at this time for June 16 at Xerox Park on the Star computer, and it'll feature a working Xerox Star. And we are able to present a working Xerox Star because Dave Curbow of Sun Microsystems has 12 Stars in his home, which he cannibalizes in order to get one running from time to time. He thinks he can get one running, one last time, for this last talk. That's June 16, Xerox Park.

OK, so perhaps we can start now with the video.

VIDEO OF U.S. AIR FORCE FILM, "In Your Defense":

One of the most dangerous threats to our nation's security is the possibility of attack by high-speed enemy bombers armed with nuclear weapons. These bombers can strike at supersonic speeds from many directions and altitudes to confuse our defense and delay the dispatch of interceptor weapons. In a mass raid, high-speed bombers could be in on us before we could determine their tracks. And then it would be too late to act. We cannot afford to take that chance.

It is to meet this threat that the Air Force has been developing SAGE, the Semi-Automatic Ground Environment System, a network of geographical defense sectors covering the continental United States and extending into Canada. Each sector has as its focal point a computer installation known as a Direction Center. A group of sectors comprise a NORAD division, where the key point is a computer installation called a Combat Center. Combat Centers are headquarters for higher echelons of command, integrating the individual sectors into a centralized defense system.

Let's suppose an enemy launched a surprise attack with high-speed aircraft, intent on destroying our great cities. The long-range radars, which constantly scan the skies, detect all planes in flight. The information is flashed electronically to the Direction Center, where the computer instantaneously makes the necessary calculations and displays its findings as tracks on the scopes of the consoles. In the Air Surveillance Section, tracking operators monitor all traffic in the sector, and quickly relay unidentified tracks to operators in the Identification Section, who determine that the tracks are hostile.

An alarm sounds in the Weapons Direction section. The Senior Director, and Senior Weapons Director, size up the threat, using all the up-to-the-minute information displayed before them on the scope. On the basis of this information, constantly supplied by the computer, the track is assigned to a Weapons Director, who decides that interceptor aircraft should be sent against the enemy. Immediately, the Weapons Director assigns the control of the intercept to an Intercept Director, whose situation display shows him that fighter craft have been scrambled. The situation display also indicates that the interceptors are assigned to track A-137. "I" indicates interceptor aircraft, "G" that tracking is good, "1" the number of the controlling Weapons Director, "2" the number of the Intercept Director. A vector symbol indicates the heading of the fighters as they fly toward the point at which interception will take place. RL-15 is the intercept flight code number.

Thus the Intercept Director has the information he needs to direct his portion of the air battle. And the battle begins. At 28,000 feet, the bombers speed toward their targets. Jet interceptors roar to the attack, guided by electronic impulses received directly from the computer, supervised by the Intercept Director, miles away. Now comes the test. Were the calculations correct? Were our judgments right? With weapons rushing toward each other faster than the speed of sound, the slightest error can mean failure, and bombers free to strike with nuclear bombs.

The console display shows the intercept weapons approaching. And the battle is joined. The fighters lock on target, and rockets are fired.

The battle is over, the enemy destroyed. But one important task yet remains for the computer: to guide the interceptors safely back to their bases.

The SAGE System provides us, on the North American continent,. the finest possible air defense against attack by manned bombers, the number one threat of today, and quite a few tomorrows.

END OF AIR FORCE VIDEO

PAUL EDWARDS:

I’m Paul Edwards. I don’t know if anybody can compete with what we just saw [laughter], so please forgive me if I’m not quite as brilliant as that.

As Peter pointed out earlier, I was far too young to have worked on this project, or even to have seen much of it in action. So as not to embarrass myself, I’m going to speak as quickly as I can and leave the details of the participant action to those who were there: Les and James. Part of what I want to do is, Les and James are going to talk about the hardware and the software of SAGE, and I will leave out many of the technical details and talk about the political context in which SAGE came into being. Which was, of course, the early part of the cold war.

SAGE actually began in 1944, before the war was over, as the Airplane Stability and Control Analyzer, which was an analog computer designed to serve as a controller for a flight simulator. Flight simulators in this period were basically little boxes that you could get into, whose attitude would change -- you’d work them with a stick, and it would simulate an airplane in that sense. The idea of the Airplane Stability and Control Analyzer was to build a general purpose flight simulator that could be programmed with the characteristics of any airplane, for the purposes of training pilots or perhaps for designing the aircraft itself, so you could see how it would respond to the controls.

The project was handed, by the Navy’s Special Devices Center, to the MIT Servomechanisms Laboratory. Jay Forrester, who was then an advanced graduate student -- he had come to MIT in 1940 and joined the Servomechanisms Lab -- took on this project as the leader. He spent about a year trying to make this analog computer work for this purpose. It’s important to realize, I think, that in this period it was not obvious that there was a better way to do it than that. Digital computers were, of course, under development, but nobody really knew that they could be made to work, to work reliably, to do the kinds of things that Forrester wanted them to do. And Forrester’s main problem with the analog techniques was not that they were inaccurate, though they were, of course, because, as you all know, analog computers have inherently limited accuracy. His main problem was that they were slow, because they were electromechanical, in those days; they were ball-and-disk or motor-drive analog machines. And for a flight simulator you need a real-time controller, because what you are testing is the response of the aircraft to the controls. This meant that the computer had to work with what was then extreme speed.

He gradually came to believe that it would be better to go with a digital technique, and around 1946 he switched to that technique after hearing about the ENIAC and other projects. Then he faced a different set of problems. One of the worst of these was the unreliability of vacuum tubes. They were expensive, and they burned out a lot, so as the computer ran, it would constantly be burning out tubes; it meant that they had to be found and replaced, and caused lots of downtime. Forrester was able to solve that problem eventually by redesigning the tubes; he actually made some major contributions to that problem.

The computer for the flight simulator had some unique features, the main ones being that it was designed for real-time operation and for control computing. Virtually all other digital computer projects at that time were calculators. They were going to be used for science: for off-line calculation of complicated mathematical problems. Forrester’s goal was much more linked with the machine he was actually trying to control. So, to achieve speed, control, and reliability, meant that this project cost a lot of money. The final total spent on the Whirlwind computer, which was a digital version of the ASCA [Airplane Stability and Control Analyzer], was about five million dollars, in 1950's dollars; that today would be about 20-40-who knows- some much higher number. Other computer projects of this period were typically $100,000 to $500,000. So, Whirlwind was 10 times as expensive as even the most expensive of most of the other projects around.

Here's Forrester. One of the inventions that he made for the SAGE project was core memory. Actually, co-invented at the same time -- around the same time -- by somebody from RCA, but this is him, with a core memory plane, and I'm sure the others will talk later about the cores that are in these boxes you see up here.

Remember that up through 1948, the idea of Whirlwind was that it was going to be a controller for a flight simulator. The Office of Naval Research, which was paying for this project, was rapidly losing interest, because they were spending millions of dollars a year for what they were now starting to see was really only another general-purpose digital computer project. At that time there were already 13 other digital computer projects, also very expensive, being funded by various agencies, and the question was, "Why shall we pay for this one?"

For the 1949 fiscal year, MIT requested 1.5 million dollars for the Whirlwind project. That would have amounted to something like 80% of the ONR mathematics program budget, and about 20% of the total amount that the ONR was spending on all contract research put together. The actual budget was $1.2 million for that year, so there was a really huge investment in this project. But it was made very clear to Forrester that if he did not produce results, in the form of a working simulator controller, very soon, he was going to lose his sponsor.

So in '48-'49 Forrester began to cast about for new sponsors, and, therefore, for new justifications for the project, since -- flight simulator -- so what? He was in a good position to do this because during the immediate period after the war he had already been looking at all kinds of applications for his machine; not just military ones, but also things like insurance calculations. He had gone into quite some depth in thinking about how digital computers could be used for fire control; that's the problem of controlling guns. He had thought about a digital computer as the centerpiece of a combat information center, the sort of telecomm center on a ship, and he had written a couple of long reports about digital computers as controllers for defense systems in naval and antisubmarine warfare.

By 1948 he had done so much of this thinking that the Whirlwind group was able to develop a plan -- a 15-year plan -- for digital computers in military operations. They projected that it would cost two billion dollars to carry out this program, and that digital computers would be applied to air traffic control, missile guidance, logistics, fire control, and, especially important here, real-time computerized command and control. This was an enormous vision, much bigger than what most people were thinking about for digital computers at this period.

Well, Forrester was lucky. He had this set of justifications more or less pre-prepared; he still needed a sponsor; and right around the time when Whirlwind was about to be killed -- the proposal was to cut its budget to about 10% of what it had been for the 1950 fiscal year -- the Soviet Union set off its first atomic bomb, in 1949. Now, here's where the context gets very interesting. The Air Force, before and during World War II, thought about air defense as not what we think of today, that is, defending against an incoming attack, but as destroying the sources of enemy fire. That's defending yourself against an air attack. Well, that could mean destroying them before they leave the ground. In fact, the initial use of this term was in the context of using airplanes to attack ships at sea, because they would be shelling the land, and you would send an airplane out to destroy them.

Up to 1949 there were no Soviet nuclear weapons. There had been a campaign for a better radar-based continental air defense in 1948, but it had failed, and the only air defense program that had been approved was a very, very minimal lash-up system, it was called, that was going to have something like 85 radar stations to cover the entire United States. Then the Soviet atomic bomb was exploded, and immediately this posture of very little air defense became a serious public relations problem for the Air Force. Civilians began to demand an active air defense, particularly those who lived in Washington State around the Hanford Nuclear Reservation, Boeing, and so on. They were not that far from the Soviet Union and they could see the problem.

At this point what began to happen was that immense amounts of money began to flow into the Air Force and the rest of the services to meet this problem. And they tried all kinds of things. They speeded up construction on the lash-up radar system. One of my favorite little stories from this period is something called the Ground Observer Corps. This was a volunteer program, begun in about this period, which consisted of setting up observation platforms, mostly along the northern border of the U.S. and up into Canada. And the Air Force had a radio ad campaign that asked people to volunteer to serve their country, to defend against incoming bombers. So it sent these people up onto the platforms where they would watch the sky with binoculars. And they saw lots of things! [Laughter] They saw airplanes -- many of them were civilian; they saw birds; they saw all kinds of things, and most of them they thought were Soviet bombers. I mean, this was a scary period. They would then telephone the nearest air base, which would then have to figure out if this information was worth anything. And pretty much none of it was worth anything. So, it very rapidly became obvious that despite the huge size of this program -- there were 8,000 observation posts, and at the peak of the program 305,000 volunteers staffing these things 24 hours a day -- the information was pretty much useless. So, commanders just ignored it. For one thing, by the time it had been verified, the bombers would already be there. So, what was the point?

The reason I'm telling you this story is that the purpose of this program was not really air defense. It was public relations. It was saying to the public, "We are doing something about this problem -- we see it." At the same time, the Air Force started looking everywhere for ideas from scientists and engineers, and [referring to the slide presentation] now we're restarting, and I'm not sure what's going on.

There was another factor here, and that was that the strategic doctrine of the 15 years or so following World War II was basically this: the best defense is a good offense. And this goes back again to that prewar period when the idea of air defense was to attack the source of enemy fire at the earliest possible point. The U.S. has a huge perimeter. Protecting the whole thing seemed like an impossible task. It would be easier, and better, to get there first. And that, in fact, was the strategy. It was known as "prompt use." This is basically a doctrine of preemptive strike. The idea was, we would be watching with every means at our disposal for movements of aircraft that looked like they might be mobilizing for a strike, and if it got far enough, we would simply bomb them before they got off the ground. For this reason, the Air Force did not actually expect to need a continental air defense. There was no point in it.

This strategy was adopted for a number of reasons. One of them was that there were very few nuclear weapons in this period; it took more than a decade for significant numbers of them to be assembled, so that using them at all meant that you had to use them before they might be destroyed here.

There's another reason for this, a strategic reason, which is the problem of air defense. The perimeter is huge, the issue of even finding the enemy bombers at all, when they might be flying under your radar, or whatever, was hard. How do you maintain coordination of a defense effort on that scale?

There was the issue of whether you could shoot them down. The best kill ratios historically recorded in air warfare were during the Battle of Britain, when about 10% of attacking planes were shot down. That worked in the Battle of Britain -- it was won -- but it wouldn't work in nuclear war, because even half a dozen airplanes getting through is still a catastrophe.

Finally, there's the problem, as we all probably remember from the cold war, of hair triggers. That is, preventing false alerts and the launch of an attack in a situation which is not actually warranted.

So, a guy named George Valley, another MIT professor, was assigned basically to this task by the Air Force, and he heard about the Whirlwind. He decided to adopt it for radar calculations in a large-scale, sort of high-tech, air defense system. He met Forrester, he saw these various reports that Forrester had already prepared for other military applications of computers, and he was impressed. He ended up rescuing the Whirlwind and using it as the basis for SAGE.

The plan for SAGE was developed at MIT during some summer studies in 1951-52. It was called something else then. It was actually called the Lincoln Transition System for a while. I think the name SAGE came in 1954. I'm taking more time than I wanted to so I'm going to skip over this quickly, but it's interesting to notice that there was in fact an analog control alternative that went on for some years; the idea was simply to automate manual calculations to some degree with calculating aids, and pretty much use the old system.

So, the project that we just saw the movie about was SAGE, the Semi-Automatic Ground Environment. It was a radar-based early warning system that used centralized, networked computer control. SAGE was deployed between 1958 and 1961. There were actually, I think, 23 sectors, and the original vacuum-tube computers -- the last one was finally taken down in 1983, still operating. The total cost of the system is hard to calculate, but it was somewhere between four and twelve billion dollars, in dollars of around 1960, which would be about five times that much today, so really, quite a bit of money.

The SAGE Direction Centers, which you saw briefly during the film, were four-story concrete blockhouses. They had their own power plants, partly because the power grid might go down in the event of a war, but mainly because the machines took so much power: they had an entire floor devoted to air conditioning. One floor was devoted to the two duplexed computers. We already heard the statistics of the enormous size of the machine.

One effect of SAGE was the enormous contract that IBM got for building the SAGE computers; they received about 500 million dollars during this period, to build the 56 computers. It was the single largest contract IBM had in the 1950's.

This is unfortunately a little hard to see, but this is a block diagram of the SAGE Direction Center. This up here, the top floor is the control center; I think this floor is the computers, and down here is air conditioning and off here to the side is the power plant and its cooling towers. Here's an interior of the SAGE control room. We'll skip through these quickly since we've already ... the monitors gave off a blue light, and they were sometimes called "blue rooms" because of this strange glow that they emitted.

SAGE was responsible for many, many technical advances, perhaps more than any single computer project: magnetic core memory I already mentioned; Forrester's highly improved vacuum tubes which were much more reliable than those that existed before; the operationalizing of duplexing -- linking the two computers together so they would share data and one could go down and the other one would continue. The total downtime of SAGE computers was somewhere between one and ten hours a year. This is in a period when most computers were down for numbers of weeks, or even months, all told. SAGE was responsible in part for digital data transmission over phone lines; for solving all kinds of problems in analog/digital conversion; the use of modems; it was, in a certain sense, a network -- all the Direction Centers were linked, and shared data about their immediate situations so the overlap problem could be solved; video displays, as you saw in the film; graphic display techniques; and, maybe most important, real-time digital control. Again, not a use most people thought of in this period.

He was also responsible for the first algebraic computer language; the first real-time software. The programs written for SAGE were enormous, by the standards of that day, and even by today's standards. The first operational program in 1958 was about a quarter of a million lines of code -- the most complex system ever built -- and one of the interesting problems faced by the SAGE designers was how to organize people to write such a program. By the early 1960's in some of the spin-offs from SAGE these systems were up to a million lines of code. James will talk about the Systems Development Corporation, I assume, but this was a spin-off of the RAND Corporation in Santa Monica that did the programming for the SAGE system.

At its peak this effort had 800 programmers, which doesn't sound like a lot by today's standards, but at that time, by some estimates, this was about 20% of all the programmers in the world. [Laughter] It was the largest programming effort of the 1950's, and it was extremely influential: many people left SDC or other SAGE programming efforts and went on to found their own companies, and spread knowledge.

This is the cover of the IBM manual for the SAGE computer.

I'm going to close by talking a little bit about SAGE as a cold war product and a cold war social project. The issue of air defense for which SAGE was developed gave the project a sense of urgency which it would not otherwise have had. That meant in turn that there was almost unlimited funding available for this, and we can wonder what would have happened had that Soviet atomic bomb never gone off. The problem it was trying to solve -- the problem of nuclear command and control -- was an extremely difficult problem. As the cold war dragged on and it became obvious that this meant basically 24-hour, 365 day a year, permanent alert, the rationale for using a computer to do this control became more and more telling. These machines had to be extremely reliable -- being down for even an hour is a problem in a situation like that. Nuclear command systems must be centralized, because of the issue of renegades -- people taking off with their own weapons. And finally, of course, the machines must operate in real time.

It's an interesting project to look at because one might think that the agency behind this effort was the Air Force. In fact, it was not. It was the group of scientists and engineers, many of them from MIT and RAND, who really pushed this project through. There was actually a great deal of resistance, especially at the higher levels of the Air Force, to SAGE for many reasons. The most fundamental reason was the strategic issue I mentioned before. Since the Air Force did not expect to need SAGE, it was hard for them to really back it. What happened was a process of mutual orientation, in which the civilian engineers educated the Air Force about the potential of these new machines, and in turn the military funders directed the engineers toward this particular solution to that problem.

SAGE was extremely successful, at least in this sense: that by the time it was over, the Air Force had been completely transformed from a group that initially resisted computerization to the leading advocate of computerization within the military. SAGE has a lot of legacies; many of them are technical, and we'll hear more about those in a minute. One of them was more computer control systems for the military: there were something like 25 other computerized command and control systems being built in the late '50's and '60's, as soon as the SAGE systems became operational. One of them is the Strategic Air Command Control System, the World-Wide Military Command and Control System, and in fact, if you look at the Strategic Defense Initiative of 1983, it's really just SAGE all over again, with a different set of weapons.

There are also some ironies about SAGE. It was almost obsolete by the time it was completed, because ICBM's made its warning abilities superfluous. It probably would not have worked: it was easily jammed; many of the tests of the system were fudged to make it appear that it would work, but it probably wouldn't have. And the Air Force, as I said, initially didn't want it.

But SAGE did work in many other ways. It worked to build a computer industry. It worked to justify air defense. It transformed the Air Force into a high-technology force. And maybe socially most important, it created what became a very long-standing belief that there was a technological solution to the problem of nuclear war. I'll stop there. [Applause.]

JAMES WONG:

My name is James Wong, and I'm going to try and tell a story about my life with SAGE. I want to first thank Dag for inviting me to speak about SAGE here. In the last three or four weeks, thinking about SAGE and trying to get a memory dump, it's like going home, really, back to SAGE. Unfortunately -- if I knew I was going to be standing here today, 20 years ago, I would not have cleaned out my attic and threw all those documents away! [Laughter] So much for being a packrat, right? I did save them for 20 years though, right? [Laughter]

Anyway, by this time you know that Lincoln Lab was the birthplace of SAGE. Lincoln Lab was chartered to be an R&D organization, so they consented to start the software development if some other organization would pick it up and do the development and, later on, installation at all the sites. The Air Force looked around to find somebody, and the only logical choice was the RAND Corporation, because RAND had a lock on the programmers in the country. RAND had about 10% of the programmers, and that was 25 experienced programmers. [Laughter] So the industry experts said at that time there were like 200 experienced programmers. So we were the logical choice.

So this was 1955, and I was very fortunate to be one of eight people as the nucleus to go back to Lincoln Lab to work with them, and the objective was to pick up the programming function, transfer it from Lincoln to RAND Corporation, and then move it out here to the west coast in Santa Monica. So that was the key. The original commitment [was] 25 programmers, and the agreement with Lincoln Lab was, we would be working together as one organization, and the RAND programmers would be integrated into the Lincoln organization.

And here we were, back on the east coast, full of vim and vigor, young, idealistic, ready to conquer the world. Remember, at that time the chill of the cold war was on, so there was a lot of anxiety at that time. And here we were, back east, out of our environment actually because we were all from the west coast. And our motivation -- what was our motivation? I guess we had quite a few. First one was, we're going to protect the country. The second one was, we're going to be doing something that nobody has ever done before. A third thing was, we want to prove the skeptics wrong -- those who said this was a Mission Impossible. And, that's our motivation for going into this thing.

I remember the first time I walked into the computer room at Lincoln Lab, and this is what I encountered: it's awesome! Cabinets upon cabinets of vacuum tubes, and I think Paul mentioned there was something like 58,000 vacuum tubes, something like that, and it's mind boggling! How can so many vacuum tubes be working without some sort of failure? You know, it's really mind boggling. And then we were told, "Hey, it's only a simplex machine here, in this lab. Out on site you're going to have a duplex computer." So you can imagine. And you walk around the cabinets, you just feel like you're engulfed in there. You get a strange feeling that you're part of the computer, and the computer is part of you! It's like a member of the family. It's sort of weird, when you get right down to it.

From there, my first assignment, since I had an electronics background, was to work with the equipment team to check out the long-range radars, the gap-fillers, the ground-to-air data link, and the height-finders. And my specific assignment was to write a test program to check out the ground-to-air data link. Two Bell Labs employees were assigned with me to work on this program, and at that time everybody really worked hard. Everybody: Lincoln Lab personnel, as well as RAND, and Burroughs people involved, IBM, and RCA. There were many, many organizations involved. So there wasn't just a few companies involved in this thing.

We worked about three weeks, and finally we got the program written, got it keypunched, and we were ready to assemble the program, and make a run. In those days, computer time was real tight, so if you want to do an attended run, you have to run from midnight to dawn. If you want unattended runs you can submit your runs and they'll run them for you during the daytime. Since the Bell Labs people and I decided -- we had like a pride of authorship in this computer program -- we wanted to see it assembled and run firsthand. So we scheduled time at midnight. We all agreed we would show up 10 minutes early; so that in case something goes wrong we still can get our time in. At the appointed day, about five minutes to twelve, I would come running into the computer room with my deck of cards, and I decided to play a little joke on my friends. I come running in the computer room, and pretend I tripped, and fowwww! All the cards were on the computer [room] floor, right? [Laughter] You should have seen my associates! They were down on the floor trying to reassemble the deck! Of course they couldn't do it, because even though the deck was sequenced, some of it, we had made changes to it and inserted cards in between. In those days, we had one card per instruction; you had the ops code, and the address, and that's what it was. After scrambling around for awhile, I went back to the back of the cabinet and brought out a duplicate deck  You should have seen the faces on those guys! [Laughter] I almost got killed for that one! [Laughter]

So, everybody worked hard, but everybody -- there were a lot of practical jokes going around, too. It made the things lighter, and just eased the tension some. But those were the days -- they were really heavy days. Schedules were tight, and everybody worked day and night. We were not getting compensated either, at that time, for working overtime hours.

My next assignment: once again I was very fortunate to be involved with this one. I was assigned to what was called the Program Executive Control, the most important program in the SAGE system. [In] today's language that would be called the operating system. The operating system, PEC it's called, was designed with a radar sweep of 15 seconds in mind. In the computer it was called one frame of data. The Executive was broken down into three sub-frames, so you have five seconds each sub-frame. Program modules would operate in the first sub-frame, a different set of program modules can operation on the second sub-frame, and the third sub-frame had a different set of modules. Lincoln Lab was very ingenious in many of their designs and their concepts, and this is one of them, this Program Executive Control.

A second one is the compool; as the name implies, it’s a common pool of data that program modules would use. They would either put data into a particular location, later on another program might use that information. So the compool was another innovation; as a matter of fact the COBOL compool was based on the SAGE system. A third innovation was called PTRS, Program Test and Recording System. Everything run in the SAGE program was recorded on tape, and the tape can be later dumped for analysis.

At that time period the biggest problem we had, since this system was a huge system, first time put together, the requirements specifications were vague and incomplete. At that time -- we all know this: if you don’t nail down requirements [laughter], if you have three programmers you’re going to have three different interpretations, right? So, that’s the problem we faced. The specifications [were] called in those days, I remember, it was opspecs and mathspecs. Those were the requirements specifications. And on top of that, they were classified, Secret or Confidential, so difficult to get to in the first place. [Laughter] So the priority at that time, before you can start writing coding specs and writing code, was to get the opspecs and mathspecs tightened up, so that everybody is on the same page. That was one of the biggest problems involved; that means schedules were slipping, and it’s an interesting thing, because all the programmers were sitting in one room trying to get these specs together, and we looked at each other and say “Who’s doing the work back at the office? There’s nobody back at the office doing anything.”

From there, putting the programs together -- many, many program modules, as I remember something like 90 to 100 program modules that take care of the radar input, the identification functions, the weapons assignment function, the switchover from computer to computer -- there were about 90 to 100 software modules.

From there, this is now getting into 1957. Did I get the dates right? Yeah. It was time now, the programs were being checked, and it was ready to move the program to McGuire [Air Force Base, New Jersey], to check out the McGuire SAGE site. I was one of the team leaders, to lead the team out there. The interesting question that came up at that time was: can we test and integrate new hardware, new software, and new procedures at the same time? Nobody had an answer. So what I called the massive manning concept was applied. [Laughter] We had 70 programmers out on site. Everybody -- people were working day and night, constantly, because computer time was in short supply. Because other organizations were using computers as well: IBM, Burroughs, RCA, and other organizations were using computer time also.

Getting out on site, now. We anticipated problems, of course, and every time a problem showed up, our programmers would say, “Hardware!” [Laughter] Hardware people would say, “Software!” [Laughter] So there was a lot of finger-pointing going on, because there was a lot of problems. [Laughter] Fortunately, at that time IBM had a very liberal awards program for the engineers: if the engineers can come up with a fix for a problem in the hardware, depending on the complexity of the problem, they would get awards. The nominal awards were like $50 and $100, and it went up to $1,000 to $15,000. I remember, as soon as that 15 [thousand] dollar prize was awarded to somebody, those engineers came over to us and said, “Would you help us isolate this problem, and develop a fix for us, and we’ll share the awards with you.” [Laughter] There was still finger-pointing, but they were friendly finger-pointing from then on! [Laughter]

This is getting to be the latter part of ‘57, now. There were some changes going on back at RAND headquarters. The System Development Division of RAND, which was responsible for SAGE, had grown so big that it was like the tail wagging the dog. And RAND had the same kind of problem that Lincoln had -- RAND was chartered also to do long-range research. So they didn’t want to do the production and installation. So they spun off the new organization, called System Development Corporation, so one day I was working for RAND, the next day I was working for SDC. So that’s the way it was. Also at the same time, toward the end of ‘57, the SAGE computer came to Santa Monica. That would be now the test-bed for development and maintenance of future software. I was fortunate enough to take the team from Lincoln back to Santa Monica; checked out the computer, and from that point on, the official handover of the software function from Lincoln to RAND was complete.

We’re going up to 1958 now. In June of 1958 McGuire went on-line. That’s just about two years after the equipment arrived on-site, at the blockhouse over there. Now Stewart Air Force Base [New York] was the next SAGE site to go on-line, and that went on in September of the same year, ‘58. They needed only 40 programmers to do the installation. Subsequent sites went on-line about every two months, until 1961, when all the SAGE sites were completed. For those sites, they needed only 15 programmers. So we went from 70 to 40 to 15. I guess in all -- I think Paul mentioned there were 20 sites -- I think we had a count of 20 Direction Center sites, four Combat Center sites, one combination Direction Center and Combat [Center] site, and that was up in North Bay, Canada -- that’s the computer that’s the combination Combat and Direction center site. As a matter of fact, I have a friend back here, Dave Burley, he was at North Bay, Ontario, and I’ll bet you he can still find his finger marks on some of those knobs! [Laughter]

Paul mentioned that SDC had really launched programmers into the computer industry, and it’s very true. At the height of -- SDC must have trained over 2,000 programmers. Paul mentioned also that we had like 400 programmers at any one time, that’s about right, the right ballpark. Half of it was at home, doing the modifications, the other half was on site supporting the installations. And of course, by that time, the other half was in industry, which is about another 800. Many of them, like Paul said, started their own companies, became managers of data processing organizations. They really cut their eyeteeth on SAGE, all these programmers, all these managers.

The word around the industry was, “It was first done in SAGE.” For many, many years those were the words said in industry. SAGE was the real-time, command and control computer-based system with capability so advanced that 40 years later, today, some of that capability can still be called state of the art. The computers today are a lot faster, and better, but we think about things like multiprocessing, real-time database management, distributed processing, time sharing, interactive displays, networking -- they were all there in SAGE.

That concludes my talk. Thank you. [Applause.]

Can I just say one more thing? Dave Burley, my friend, found this [document] in his files, so if anybody wants to come up and take a look -- it's the data flow in the computer system. So if anybody wants to come up and take a look at it, they're welcome to, and I'm sure Dave can answer a lot of their questions. Dave has also generously -- he's going to contribute this to the museum. [Applause.]

Q. [Inaudible]

JAMES WONG: How does anything else like this sneak into industry, right? [Laughter] I think we've all heard about the other world, and all that information has leaked out to industry [inaudible].

Q. In your talk you discussed the fact that when you started off you had loose specifications, and you had the hardware people pointing at the software people, and software people pointing at the hardware people. I'm glad to see that after 40 years, nothing's changed! [Laughter] [Someone else: We haven't learned too much, have we?]

LES EARNEST:

Hi. I sat down this afternoon to write up some of the war stories just to focus. I called this SAGE a marvelous technological solution to the wrong problem. There are many more war stories than we can have time for this evening. But let me go over a bit of it.

I admired the MIT computer work from afar, initially. When I got out of Cal Tech in 1953 I knew that I would have to go into the military in some way, because they had tried to draft me already. I found that I was uniquely qualified for a special program in the Navy: I had an engineering degree and poor eyesight. If I had been lacking either of those I couldn't have qualified, but this somehow got me into the Aeronautical Computer Laboratory, in Johnsville, Pa., where we had the world's largest and fastest computer of that era. It was called Typhoon. It was an electronic analog machine, built by RCA, and it needed check solutions, so my responsibility, as digital computer project officer, was to generate check solutions on an IBM CPC, an electromechanical computer that was a real bear to get anything to work on. It only had 40 words of storage, which was electromechanical, and you had to give it exercise in the morning to get the oil flowing, or it wouldn't work. [Laughter]

While there, I began to hear about this Navy-supported project at MIT that produced Whirlwind. What would happen was, the admirals would go to MIT, would listen to what they were told by Forrester and others, took careful notes, but they didn't quite understand what this was all about. So then they would come to our lab, and would talk to my boss, Commander B., ask him what this means. He would then at lunch time come to talk to me, and ask me. I would give it back to him, he would give it back to the admirals. [Laughter] Over time I began to find ... it was really whiz-bang stuff. I was intrigued by all of this.

So when it came time to escape from the Navy, I applied to several places. I was offered a couple of jobs at IBM, but I chose to go to MIT Lincoln Lab, and somehow I ended up in the Weapons Integration Group, which was responsible for making manned interceptors and missiles fly under SAGE control. We had to develop the equations for guidance, and we designed the Intercept Director's console, and that sort of thing.

It was all great fun. James mentioned that he did the data-link. Initially, commands to the interceptors were sent by voice radio. You'd tell them which direction and how fast and at what altitude to fly. This data-link of course automated that, so the computer could give the directions on a little display in the cockpit. The specs for that were all very carefully worked out, to show how many bits for each function, and the identification code, and all of that. I sort of expected you to mention a small problem that arose: there was one contractor that build the transmitter system, and another that built the receiver, and whereas they all agreed about how many bits for what, they didn't agree on -- there was no spec on which was sent first, the high-order bit or the low-order bit. [Laughter] Sure enough, Murphy's Law got 'em, so they got to redesign one of the systems, I forget which one. [Laughter]

One of our early tasks was to bring the Bomarc missile into SAGE control. Bomarc was a rocket-assisted takeoff; it took off vertically, would get up to speed, then would be powered by a ramjet, would heel over, fly at high altitude with a Doppler radar looking down. Presumably, when you told it when to look, it would find a bomber down below and would dive and kill it. That was in theory. [Laughter]

Well, it was a pretty complicated beast, and of course it was necessary to test the electronics of the missile when it's sitting on the ground. So they had a test mode for the missile complex, in which you throw a switch to test, and then you can send simulated firing commands to the various missiles and you check whether they received them and understood, and all that. But of course it doesn't cause the missile to fire, it just is testing the electronics. Well, when we were asked to review this, one of our engineers looked very carefully at this system, and discovered that if you send the missiles the test commands, and then throw the switch from Test to Operate, without individually resetting each missile, they will all erect and fire. [Laughter] As soon as we mentioned this to Boeing, they quickly concocted a fix for that.

A year or so later I somehow was selected to lead the team that obtained nuclear warhead certification for the Bomarc missile, so we had to go through all the hardware and software and prove that the probability of failure -- combination failure of anything -- would produce an inadvertent launching, with probability less than 10-to-the-minus-very-large-number on any given day, and that it would take at least two berserk people to do it by themselves; that is, one person couldn't do a launch.

Well, we were able to convince ourselves eventually that it was okay, but along the way we discovered a rather frightening thing about the way it worked. All the telephone lines from the computer out to the launch sites were duplexed for reliability, so if one goes down you have another way of issuing the launch commands. All of SAGE was built that way, with backup and redundancy. So, at the far end of the line, there's a little black box that listens to the primary line, and if it senses that that has gone bad, it automatically switches to the backup. Unfortunately, we discovered that if the backup is also bad, what it does is amplifies the noise and generates a random bit stream. [Laughter] So one of our guys did a Markov analysis to figure out how long it would take to get a firing command; it turned out to be a little over two minutes. [Laughter] However, in order for the missile to actually launch, it has to get a full set of commands -- that is, direction, speed, altitude, in addition to the firing command. And it had to get this within a certain length of time. We were able to show that the probability of getting all of that stuff within the required time frame was very, very small, very unlikely. Therefore the effect of this would be, the missile would erect, and then abort. I published an analysis of this with the provocative title "Inadvertent Erection of the IM-99A." [Laughter] As luck would have it, two weeks after we released the report, it happened, in suburban Washington, D.C. [Laughter] So, I promptly found myself to be the head of a committee to fix that, and it was very easy. You just sensed bad line #2 and then you shut down.

This was all heady stuff, and we were able to make the hardware work pretty well, but of course when you backed off and looked at the performance specs, or the lack thereof, you realized that this was in fact a gigantic boondoggle, because there were no performance specs in the sense of an air defense system -- that function. There were literally no requirements of being able to meet a manned bomber threat of any sort. Paul mentioned the jamming. That would kill the system in an instant. It simply could not track bombers that were dropping chaff and actively jamming. There were later some hacks at trying to make it work in that kind of environment, but they were just that. And a much simpler system could have been used in that kind of environment.

If that weren't enough, however, there were certain other things that would have contributed to its downfall in a real war. It was basically a peacetime defense system -- it worked fine in peacetime. [Laughter] MIT had recommended from the beginning that these big computer facilities be buried underground so they would be reasonably well protected. The bean-counters in the Pentagon, though, saw that that was a big expense, to put the stuff underground, so they said, "Well, let's just put it in a concrete blockhouse." Then it becomes a lot more vulnerable.

But the killer on all of this was that the Air Defense Command wanted a good lifestyle; they knew that their officers would have to spend a lot of time in these facilities. They looked around at where there are good facilities. At that time, General LeMay of the Strategic Air Command had the best of everything. He had the best officers' clubs, everything was topnotch for SAC. So they put most of these computer facilities or a large fraction of them at SAC bases, where they all became bonus targets. [Laughter] Obviously, if there was a manned bomber attack, they'd go for the SAC bases first, and they'd get the air defense system as a byproduct. [Laughter] So it was not a wonderful air defense system.

In the same era, the Air Force intelligence was producing claims that the Soviet Union was developing an air defense system of a similar sort, but somehow it never materialized. I don't know whether that was because they really weren't up to it, or whether they really understood better than we did what it would take.

But then there was this wonderful facility in SAGE, the display room with glimmering blue consoles and subdued light. And it became a showcase for both the brass, and all the military, and also congressmen. Everyone decided that this is the way to run a war. This gave rise to this whole industry of L-systems, so-called, 438-L, 465-L, all this stuff, that were competing for funds with other things. And for the most part, none of them worked worth a damn. There were a few exceptions: the satellite control system, I would say, was very effective, and provided good intelligence. But most of these others, including the Strategic Defense Initiative, were basically boondoggles, and we ain't through it yet! This stuff is still going on. It gave rise to a multi-billion dollar industry that basically produces useless junk. And here we are, 40 years later, still at it. [Comment: talking about it.] Right. No -- still doing it, the government's still doing it.

Anyway, once over dimly. Thanks. [Applause.]

PETER NURKSE:

What we'll do now is have the three members of the panel take the seats here and -- why don't you use this microphone, and pass it between you as you [inaudible] as you answer questions. And if you repeat the audience's questions -- you in the audience can remind the panel [inaudible] otherwise your questions won't be recorded.

Q. Aside from assembly language, what was the other language used to program the computer?

JAMES WONG: Basically that was it, assembly language. Later on, late, they tried to put JOVIAL -- tried JOVIAL as the language. Anybody know about JOVIAL? It's a higher-level language, but it was not cost-effective and it was never put in.

LES EARNEST: Let me just comment. It might be mentioned that this was a fairly peculiar computer in some ways. It was schizophrenic: it had a split accumulator that was thought of as being x-y coordinates for geographic things, and indeed it worked quite well on that sort of thing. It turns out, though, that most of the computation was not of that nature, and therefore this structure, the architecture, didn't make a whole lot of sense.

Q. Did it have MMX too? [Laughter]

JAMES WONG: I guess that I didn't mention that this was a 32-bit machine. The left half was the ops code, and the right half was the address. So that was basically it.

Q. How much memory?

JAMES WONG: That's right. [LES EARNEXT: 64K.] Yes, it's 69K actually, they added on -- that box right there I believe is only 32K; there were two boxes, and they added another 4K later on, so it was 69K. Actually, this system is really a drum system; there are 12 drums. If you go back there you can see them back there -- there are a few back there now. The capacity was 150K words. Can you imagine? Today it's what, gigabytes? But in those days, that was really something.

Q. What was the cycle time?

LES EARNEST: Six microseconds.

Q. Did any of these concepts get into the FAA air traffic control [inaudible]?

LES EARNEST: Well, yes. Mitre, the spinoff from MIT, did get into the FAA system development business about five years after SAGE went operational, and produced some of the kluges that are now breaking down, I guess.

JAMES WONG: In addition to this, IBM was the prime contractor for the FAA.

PAUL EDWARDS: I just wanted to mention one other very important spinoff of SAGE, was the SABRE airline reservation system. SAGE stands for Semi-Automatic Ground Environment; SABRE stood originally for Semi-Automatic Business Research Environment. [Laughter]

Q. [Inaudible]

PAUL EDWARDS: The question is, how the source code was stored.

JAMES WONG: The program was actually on mag tape. The mag tape, when you started up, is read into the computer, and it's stored on drum. Except for the Executive, and a few of the programs that you need, that would be in core.

Q. [Inaudible]

PAUL EDWARDS: How are different versions ...

JAMES WONG: Versions of the software? [Inaudible] That is called the modification, right? And design change control was a huge problem in those days. Of course, software programs are never finished. It goes on and on and on and on. And, yes, what happens is, if there's a fix that had to be made on site, the site programmers would make the fix, but they were not allowed to mess with the master tape that came from home base. But they could actually run their changes, and then submit the change back to home base so they can integrate it into the whole system. That's a continuing process. If it's not an emergency fix, they would accumulate a whole bunch of fixes and they would then produce a new mod of the software and that would go out to the sites.

LES EARNEST: That was typically many months between, as I recall.

JAMES WONG: Yes, that's true.

Q. What was the prevailing view of yourselves, back then, about the pace of technology? Did you get any [inaudible] would be obsolete? [Inaudible] back then?

JAMES WONG: I guess you're saying, how did we view...

Q. Did you think by the time you worked on SAGE -- did you think that by the time [inaudible] in all 22 centers [inaudible] incredibly obsolete, much better stuff, or did you think this would be state of the art for a long time, or...

JAMES WONG: You have to remember: we were young, and innocent [laughter], and we didn't know what we were doing! We were really flying by the seat of our pants. We were just working to get the system working, and there was not enough time to think ahead, to see what would happen in a few years. That was basically it.

LES EARNEST: I came into Lincoln Lab in a somewhat different mode than most of their employees: I actually knew how to program. I think I may have been the only person they ever hired who knew that. My office-mate initially was a specialist in radar data -- he called it radah dater, being from Boston. I asked him -- I'd been an aviation electronics officer for three years, and I knew a little bit about electronic countermeasures, so I asked him, "Well, what do you do if they jam?" He said, "Well, we shut down." [Laughter] I knew -- this was my first week on the job. I knew from that point on that this was a fake. Somehow I stuck with it for five years or so, before I wandered off. It was a somewhat disillusioning experience, I would say.

Q. Since we hear that there was some number of thousands of vacuum tubes, and another certain thousand transistors, why were you using vacuum tubes, if you had transistors?

LES EARNEST: We didn't have transistors initially. And they certainly weren't suitable for all of the applications. For example, the core memory, you needed big, humongous drivers to run the core. Tubes were it, in that era. Later, the beefiness came along.

Q. [Inaudible]

PAUL EDWARDS: I think that figure probably comes from the fact that there was going to be a second generation SAGE computer that was transistorized, and one system was built and installed. But at that point it became very expensive and they abandoned it.

Q. You were talking about -- Les was --the Russians maybe doing something. Do we know anything more about, not only what they did, but did we have allies who were aware of what we were doing, and wanted a similar system in Europe, or -- how global was this kind of program?

LES EARNEST: As I remarked, I think that was really intelligence propaganda. That is, the Air Force -- as you know, the services' intelligence agencies often perform to meet the objectives of the organization. It might be mentioned also that in another sense SAGE was really the Air Force's response to Nike, another air defense system being developed by the Army. The Air

Force figured they had to outdo them, and they succeeded in that Nike eventually was integrated into SAGE. That is, SAGE became the dominant partner, even though it didn't work.

PAUL EDWARDS: That's an interesting question, and Les's response is right on. There were a lot of contests within the military about who should handle air defense, and it was again this issue of, what is this problem -- are we waiting until they get here, and then shooting them down, or are we trying to do something further out, before they actually reach U.S. soil? So the Army response was, air defense is like artillery: we shoot them as they come in. There were systems like this built for NATO, and several other NATO countries adopted similar systems. I don't know very much about the Russian air defense system, but my impression of it from the only source I found on that subject, was that it was extremely decentralized. Which is a very interesting contrast. Here in the U.S. we have a decentralized society and an extremely centralized air defense system; the Soviet Union has one of the most centralized governments in the world, and ends up with the most decentralized air defense system, in which basically everybody shoots at whatever is coming in when they see it. [Laughter]

Q. How did you debug [inaudible] tools, debugger, [inaudible]?

JAMES WONG: Basically we didn't have any tools; it was just running your program and see if it runs. If it doesn't run, you go back and sit at your desk and do it again.

Q. [Inaudible]

JAMES WONG: Yeah, it halts, or you get the wrong answer -- that type of thing. As a matter of fact I have a good story to tell you about that one. Working with programmers, way back in RAND, I learned a lesson. Any time a programmer tells you, "This program's gonna run 100% this time," bet him a dollar. You're going to win every time! [Laughter]

Q. Did you do your program development on the SAGE system itself?

JAMES WONG: The development was done on what was called the XD-1; that was the prototype of the SAGE Q-7, out on site. There were actually two development centers: one was at Lincoln Lab, with the XD-1, and the other one was at Kingston, the XD-2. So we used both those sites for development.

LES EARNEST: The XD-1 was a simplex system, wasn't it? [JAMES WONG: Yes.] That is, there was not duplex, to switch.

Q. This [inaudible] here, wasn't there something about Polaroid cameras being used for debugging? That was a story that was told to me by an SDC employee, that they had Polaroid cameras -- the story I heard was that they had a Polaroid camera mounted in front of the console, and if the system crashed they would take a picture of the lights on the console and give that....

JAMES WONG: Yeah, you're right, 100% correct. That was one of the things, that's it as a matter of fact.

Q. When the Nike system was, quote, integrated into the system, what had to be changed? Were there software changes, or hardware changes, or ...?

LES EARNEST: It was mostly software. Of course there had to be a digital link connecting the Nike site to the SAGE site. But that was minor, compared with the software that had to be done to hand over targets to them and lay it on them.

Q. Did this system have the ability to launch without human intervention, or was it always ....

LES EARNEST: No. No, always required two levels -- the question was, was the system able to launch automatically, that is, without human intervention. No, the Weapons Director had to designate the target, hand it to an Intercept Director, who would then launch a missile or a manned interceptor to get it.

PAUL EDWARDS: That was, however, contemplated by the designers.

Q. Was BMEWS ever integrated with the SAGE?

PAUL EDWARDS: Was BMEWS ever integrated with SAGE? Yes, I think.

LES EARNEST: The information flow, yes. It was not automatic.

Q. [Inaudible]

JAMES WONG: Yeah, the radar data came in from the DEW Line into the Direction Centers located in the northern part of the U.S. The radar data actually came in from the DEW Line.

Q. What sort of data rates were these land links, for the phone lines?

LES EARNEST: I don’t have a clear recollection.

Q. I wonder if you could confirm, or say this story is garbage, that I heard, that at some point they tried directing manned interceptors directly from the SAGE computers, and for the first test, it told the plane to make a 90-degree turn, and from then on the pilots would never turn that on again?

LES EARNEST: I don’t recall hearing the story in just that form. The initial -- excuse me, the question, he heard that the computer told the pilot to do a 90-degree turn instantly, and they realized they couldn’t do it. Well, it is true that the initial versions of the guidance program did not take into account the turning radius of the interceptors. They would, of course, servo into about the right direction over time, but there was an improved version of the software that was developed that took into account the minimum turning radius of the interceptor to get it into position. If you’re planning to make a broadside attack, which is one of the kind where you want to come in at an angle off the beam of the bomber, you have to come to a certain position, do a turn and then come in. That obviously cannot be done on a dime, so this improved version that took into account the turning radius did a better job.

Q. [Inaudible]

LES EARNEST: Oh yes. Well, what do you mean, fly it? [Q. Without the pilot.] No.

Q. [Inaudible]

LES EARNEST: Well, the switchover I think was generally -- sorry, the question is, how was the switchover made if one machine goes down. That was done generally not automatically; it was done manually. It was initiated manually.

Q. [Inaudible]

LES EARNEST: Mean time between failure? Oh boy -- I have no recollection. Probably days -- a couple of days, probably, I would guess. They did preventive maintenance on all of these tubes, of course, and would run margins on them to find those that are getting close to failure. I remember, at a certain point they were running these margins on a tube and discovered that it was absolutely dead, but the diagnostics hadn't discovered it. They traced the circuits back and discovered that during the redesign process this had been cut out of everything, so it didn't really matter. [Laughter]

JAMES WONG: Paul mentioned a little bit about the reliability of the system. The way it works is, the duplex computer -- if one fails, the other one would cover, and if both failed, the other sites would expand their coverage to cover the site that's not covered. So, with that in mind, the system itself -- the figures I heard was a 98% reliability for the system. That was totally unheard of in those days.

Q. May I ask a hardware question [remainder inaudible]

LES EARNEST: It was parallel, I believe, yeah.

Q. That would make it one of the first machines to be parallel?

LES EARNEST: Right, and basically it taught IBM how to make core memory, which led them quite a long way ahead of their competition.

Q. Can you tell us any idea, or do you have any recollection, of the type of tube modules behind you -- what kind of tubes were used in the machine [inaudible]?

LES EARNEST: It's been so long, I'm afraid I have no recollection. Now, peek! They're all there.

Q. You know, some of those tubes are very valuable for audio. [Laughter]  Lock the warehouse! [inaudible] Don't laugh -- the 5692 was widely used in SAGE computers, or so I've heard. [Inaudible] The 5692's in Japan now sell for about $250 apiece. [Inaudible]

Q. Can I ask a question? [Inaudible] pretty good games?

JAMES WONG: There were no games as such played on it, but it turned out that SDC also had a contract with the Air Force to produce simulated exercises, so they did have training for the operation of the system.

PAUL EDWARDS: And actually it was on the way to one of those simulation exercises that Allen Newell and Herbert Simon had their first conversation about building a chess computer.

LES EARNEST: There was a certain amount of horsing around, of course, in the programming classes.  I remember one fellow in my class, who went on to fame, produced a line drawing -- a vector drawing -- of a Soviet bomber that flew across the screen. Unfortunately he neglected to deal with what happens after it flies off the edge of the screen. The display space was sort of toroidal, and the bomber broke up and various pieces would drift across in different orientations. For me, I chose as an exercise to turn on all the alarms in the command center. There were about 50 of them, I think. However, although the computer can turn them on, it can't turn them off. [Laughter] So that caused a bit of consternation.

Q. I have one comment about [inaudible]

Les Earnest: I heard on the Internet that somebody developed a playmate display in which you could use a light gun to remove articles of clothing.

Q. [Inaudible]

LES EARNEST: I believe that was invented by somebody associated with Whirlwind. It was in use on the Cape Cod system, which is the precursor. Oh, sorry. Question: who invented the light pen, and I'm not sure. It was pervasive in MIT; it showed up on all the machines there. I'm not really sure who did the first one.

Q. [Inaudible]

LES EARNEST: Yeah, they did run exercises frequently, and carefully programmed them so the system wouldn't break down. Is that the answer to your ...

Q. [Inaudible]

LES EARNEST: The question is, how effective was the system? You mean, in a peacetime defense system? Yeah, it worked fine as a peacetime defense system.

Q. [Inaudible] a big bluff.

LES EARNEST: Yes, however, right, that argument has been advanced, that we bluffed the Russians into not attacking. We probably could have done it for a lot less by using papier-mache'. [Laughter].

Q. Was the choice of a light gun, as opposed to a light pen, did that have to do with the amount of electronics that had to go into it, or was it sort of the military way?

LES EARNEST: The question, was the choice of the light gun a military decision or a technological one. It was technological. It took at that time a rather complicated photo sensor that I don't think would fit in a light pen initially. Just a few years later they developed light pens that were very much smaller. Matter of fact, I used one on TX-2 to develop a cursive writing recognizer, in 1961, and it worked quite well.

Have we exhausted the supply [of questions]?

PETER NURKSE: There are refreshments. [Applause.] How about also, one last round of applause for Dag Spicer of the Museum who organized it all. [Applause]

END OF VIDEOTAPE.

Transcribed by John Amos, volunteer for The Computer Museum History Center
San Antonio, Texas    September, 1998