NASA JOHNSON SPACE CENTER ORAL HISTORY PROJECT
ORAL HISTORY 2 TRANSCRIPT

JOHN R. GARMAN
INTERVIEWED BY KEVIN M. RUSNAK
HOUSTON, TEXAS – 5 APRIL 2001

RUSNAK: Today is April 5, 2001. This interview with Jack Garman is being conducted in the offices of the Signal Corporation in Houston, Texas, for the Johnson Space Center Oral History Project. The interviewer is Kevin Rusnak, assisted by Carol Butler and Sandra Johnson.

I'd like to thank you once again for coming out this morning.

GARMAN: Sure. Thank you.

RUSNAK: I guess we can pick up where we left off last time. We had just gotten the Shuttle flying and you had talked about the computers there. So let's see where this took your career.

GARMAN: As I recall, I think I had mentioned that I was standing in the big computer room one day and was sort of touching the computers, and I concluded that I needed to move on to something else because I'd seen too many people get attached to what they were running, as if it was their empire. Maybe it was my empire, who knows. But I remember thinking at that moment that it was time to move on.

At that time, John [W.] Aaron, who I'm sure you've talked to—he was the EECOM [Electrical, Environmental, and Communications Engineer] on Apollo 13 and he'd grown to be the division chief for Spacecraft Software Division, which was what I was in. We did the software for the onboard computers in the Shuttle. He was the division chief, I was the deputy division chief, and I said, "I think I'd like to go do something else for a while."

So they agreed to move me up to the Directorate Office. I don't remember what the title was. But to start thinking about the future programs. The timing was right. It was soon after the first Shuttle flight, and I had remembered after Apollo I wanted to get moving on to Shuttle, so I said, "Heck, I don't know what's coming next, but why don't I get on?" So I did.

Station was still four years away from anything formal happening. This was '81 and '82, somewhere in there. I don't think it was till the mid-eighties that Ronald Reagan finally said, "Let's go do the Space Station." But NASA had a program of RTOPs, Research Technology Operating Plan. We called the function by the name of the form, RTOP. It's NASA's usual thing.

So what NASA had been doing was it had been doing some of these research programs, RTOP programs, to pave the way or to get ready to do Station, without it being a formal program. That sounded good to me, particularly on the software engineering side. I had dived into that HAL-S [High Order Assembly Language/Shuttle] programming activity after Apollo, going into Shuttle, and we'd made this huge investment in this Software Production Facility. So I said, "Maybe I ought to work on that for Station."

About this time, the Department of Defense had started working on software engineering, too. Its software was becoming a big pain for everyone. This was the early days of what later became known as ADA, the big programming language. I don't remember where it was born, but the notion of creating for Space Station a software engineering environment, I think the phrase we used was SSE. You know, it's embarrassing, I can't even remember what it stands for. Support Software Environment, or something like that. It ended up being a rather large contract, as a matter of fact. But the notion was that on Station we were going to have computers in the Space Station that had software that was built all over the world because of all the international partners that were going to be involved, as well as principal investigators and what have you.

We knew that it was pretty expensive to bring people to a big Software Production Facility to do tests and verification and all that. So what if we built something that could be put in a smaller computer that replicated what would be in the large central ones for simulation and ship it out to people like a kit. That was the notion. And to move to the next phase, current technology, if you will, for software engineering.

At the same time, I might add that Intermetrics, now called Averstar, the company that built the HAL-S program—let's see, it was Fred Martin and Dan Lakely [phonetic]—but Fred and Dan were still, they were quite active in pushing their company to do higher-order languages in software engineering. The HAL language became one of the contenders for ADA in DoD and I remember getting involved in that a little bit, too.

The relations between NASA and DoD were always cordial, but never really close and I don't think HAL, the language, ever really had a chance, but it was seriously discussed and I remember some conferences or something that I ended up going to in that timeframe.

At any rate, for a couple of years there, I remember being the only person in that directorate. What were we called then? Data Systems and Analysis Directorate, DSAD. Bill [Howard W.] Tindall was the director, Lyn [Lynwood] Dunseith was the deputy director. Those were my two mentors, so I had no problem moving the director's office.

I remember that during that period I was the only person working on Station in that directorate. If we're in the early stages, where we're just doing RTOP kind of work, that always comes out of Engineering Directorate. I was in a directorate that was mostly operations, and of course, the early Shuttle flights were still flying. We're still in the DDT—design, development, test—engineering phase of Shuttle, the test flights, I think they call them, the first five flights. So there was a lot of focus on making Shuttle work, too. So I was a little bit of a Lewis and Clark from the outside, exploring the unknown.

I remember, it was odd because there was almost a Romeo and Juliet thing going on there, not in the context of Shakespeare at all, but I mean because the Engineering Directorate was doing all the work and there's always a lot of competition between directorates for funds, this was one of those rare times where I was having so much fun doing this, that the Engineering Directorate kind of adopted me. They actually put me in to chair some committees and do some stuff like that, and I wasn't even from Engineering. That was great fun, and I think it really did help get a lot of cooperation between the two directorates. Hence, the Romeo and Juliet analog there.

I forget exactly what happened when. There was an organizational change. I think it was soon after Gerry [Gerald D.] Griffin came in as the center director. He restructured the center a bit and a fellow named Jerry [C.] Bostick became head of the directorate. I really don't remember, but the opportunity to go run one of the divisions, the institutional computing, I forget what they called it beforehand, but we renamed it the Data Processing Systems Division when I took it, but I ended up taking that job and moved out of the Directorate Office to run the division.

I'd become so embedded in the Station Program that they sent some of that work with me, particularly the SSE project, and that gave me the opportunity to grab a couple of other people. One of them that would be very worthwhile chatting with is Jim [James L.] Raney. Jim was the guy that built the ISAGC, the Interpretively Simulated Apollo Guidance Computer.

This was the program that ran in the UNIVAC mainframes in the Apollo crew trainers, wherein I mentioned that we actually loaded the listings of the Apollo guidance computers and it executed the listings interpretively. Hence, Interpretively Simulated Apollo Guidance Computer, or AGC. He was quite a hack, for those days, and he became quite interested in the SSE. We grabbed another fellow who I'd known for years, named Steve [Stephen A.] Gorman. He'd been in the Apollo Guidance Program Section with me when I first came on board.

Anyway, we got hold of those two guys and they picked up the SSE, but we moved it into this Data Processing Systems Division, which otherwise was doing all institutional work. It ended up doing PCs [personal computers] soon after I got there, which I'll talk about in a second. It was sort of an odd mix, because this was a leading—a bleeding-edge mission kind of thing, mixed in with the institutional computers that did everything from payroll and accounting to engineering batch runs and stuff like that.

The other thing that had happened in computing at the center, though, was that, now that I think about it, that helped draw all this together, was that John Aaron and I—I've got to get my head on straight. Dick [Richard P.] Parten was the deputy director during this early period. He'd been the division chief in spacecraft software, and Dick was very interested in computerizing the center.

The PCs weren't really around yet and we had a tremendous amount of IBM here at the Johnson Space Center in those days, because the onboard computers for Shuttle were all IBM, the Mission Control Center was all IBM, and the largest computer complex at the center was this Software Production Facility and it was IBM. So we had put the mainframe version of e-mail, called PROSS [phonetic], onto the Software Production Facility and had started using it. This is twenty years ago. There wasn't a lot of e-mail twenty years ago, trust me. It's hard to put yourself back, because today if you're not handling fifty or sixty emails a day, you're not really a person. I suppose years later somebody looking at this will say, "Well, he's crazy. We're handling a thousand a day," or something. I don't know. There wasn't a lot of e-mail back in those days, certainly not for business. There was a lot of ARPANET [Advanced Research Projects Agency Network], a lot of the research network stuff, and there was e-mail there, but it was not at all the way you think of it today.

PROSS, that mainframe IBM product, was the first real integrated package. It had calendars and e-mail and all that kind of stuff on it, real office automation. But to get to it, you needed a terminal. Well, a terminal's a keyboard and a CRT [cathode ray tube]. They don't look a lot different than PCs. We had, late in the Shuttle Program, converted everybody to terminals. Believe it or not, a lot of the Shuttle onboard software was built with card decks. But in converting the terminals, terminals started appearing everywhere.

So we started proliferating, at least around the directorate and elsewhere, all these networks, tying the terminals to the mainframes, running PROSS, which sort of moved the mission systems, the stuff building the onboard computer software, the SPF, the Software Production Facility, into the institutional world because more and more of us were using e-mail.

You know, when a good idea pops up in one organization, it tends to rise up. The leaders want it, too. And then it waterfalls back down into other organizations. So that was part of it, because in that time frame when I went to Building 12, I took all that operation with me. In other words, we consolidated all of the computer operations, including what had been this mission thing called the Software Production Facility, pulled it all together, and in that time frame, sort of sanctioned the use of PROSS and it became a center-wide system.

Fast forwarding a bit, it wasn't too long after that, that PCs starting proliferating and we were able to put boards in the PCs that made them emulate a terminal, so we were able to do the PROSS right on into the PC era. PROSS lasted from the mid-eighties all the way up until '93 or so. It was the longest single program product I've ever seen last [unclear]. One can say they use Microsoft Exchange and it's been around for a long time, but it's gone through significant changes and name changes. I'm talking about one that's very stable, very stable for years. So that was amazing.

So anyway, part of my moving into this Data Processing Systems Division was consolidating all these systems under one division. That worked well for a while, but some of the key personnel involved, another fellow, Elric [N.] McHenry was very involved. He's still at the center. He had ended up running the SPF, and so was with me in this division for a little while.

But very soon after that, they grabbed John Aaron to run the Space Station Program, to run, I guess it was Freedom, right? Yes. It was Freedom in those—it was the first—no, no, it was before Freedom. This was the first Space Station Program. Freedom was the one I did up in Washington. This was before Challenger, the pre-Challenger version of Station.

So Elric was grabbed back to be head of Spacecraft Software Division, is what happened, Elric McHenry. But his chronology and history in both the Shuttle and Station Program, if you dig deep, is another really good one to get a hold of.

Anyway, one of the major things that happened quickly was the Space Station Program that formed. John was made head of the program office at the center, and they had levels. Level A was the [NASA] Headquarters office, Level B was program, Level C was—each center was spread across the centers, and JSC was lead center, as it is now, or basically is now. All this again before Challenger.

I remember there was a lot of friction right at JSC between the Level B and the Level C office. We used to jokingly call it the BC [unclear]. There was nothing ill-intended, of course, but, nonetheless, there was quite a bit. My history of that gets dim, because I had at this point dived into running this institutional division and was moving away from the Space Station Program for a while.

I think the most profound thing that happened here was—and as I recall, this was in the spring of '84—with the startup of the Space Station Program Office, there was money to outfit these folks and they decided they wanted to be outfitted with PCs. To this point, desktop computers, PCs, had been bought under the table, around the edge. Government procurement was really tough on computers. Very difficult.

And I remember they were bought as—called calculators. They were called typewriters, all sorts of strange ways to get them. There were not a lot of computers unless they were associated with a mission or a particular project, engineering work, not office automation style, as is what everybody's into today.

So the Space Station Program was the first time there was a major buy and, of course, the Engineering Directorate, which supplied most of the people for the program office, that did most of the work, wanted to do that work. We had a little argument, and I ended up doing the work out of this institutional division, Data Processing Systems Division.

That was an amazing time, because we went off and bought hundreds of computers and started delivering them. This had really not been done before, to buy computers, start delivering them, hook them up. I mean, there were some just incredible stories, like buying bunches of computers and plugging them in and leaving and then realizing that we'd forgotten to buy printers. The joke, "You mean you want to use it, too? Here's a computer. You mean you want to use it, too?" It was pretty sad. I mean, it was lots of amateur kind of stuff. The customers, people wanting them, didn't know what they wanted.

Macs [Apple Macintosh computers] were out then, but brand new. The Mac came out in '84, January or February of '84, and that was too new to be the one. Even so, the word processing was very primitive yet. Lotus was around. That was the killer application of PCs, the spreadsheet. It's why IBM PCs became dominant in the accounting world and business world and Macs did not.

Anyway, we delivered hundreds and hundreds of machines, and that was really the beginning of the proliferation of computers at the center. Bear in mind, we already had the infrastructure. Because of the mainframe and PROSS and terminals, we had lots of networking around the center. We had what today would be called big servers, but they were mainframes in those days. And as expensive as that stuff was, that was the going rate. It was expected that you spend a lot of money on mainframes, and the incremental cost of keeping PROSS going for a larger population was not that big, so it really went along.

At the same time, the Engineering Directorate discovered DEC [Digital Equipment Corporation] machines, and that was then an extremely popular series of minicomputers. In today's lingo it would be a good desktop computer, but they were quite large. But you didn't have to have raised flooring and lots of systems programmers and pay IBM lots of money for those; you could kind of do them yourself.

So two competing systems started proliferating around the center. One was the DEC Vax system of computers, with a noncompatible network, and the other was IBM, with its very popular, but likewise not compatible with DEC network. The DEC network was called DEC Net, and it was kind of like the TCP/IP [Transmission Control Protocol/Internet Protocol] networks of today and that were popular even then in the UNIX world with ARPANET, but none of that stuff talked to each other.

When you hooked computers together in those days, it was like building private roads. You ran wires from building to building, depending on what brand of computer you had, and there was no real connectivity. It was kind of messy and created a lot of argument, as you might suspect.

It was interesting, because in my own career I sort of dived in in the middle of that stuff, got it started. This was, as I say, in the spring of '84. In January of '86, of course, Challenger happened and that just sort of stopped the clock everywhere and everything turned upside down. After the Challenger tragedy, one of the conclusions was that it was time to move the Space Station Program Office to Headquarters, that part of the issue was that the centers were too competitive and there wasn't enough communication to the lead center being located at Johnson. So let's move the lead center function up to NASA Headquarters.

I remember, clearly, John—now, bear in mind, I'd been like Rip Van Winkle. I'd been sleeping, with respect to the programs, not with respect to my job. Oh, another twist in here. I was only in the Data Processing Systems Division for a year or so and they ended up moving me up to be deputy director of that directorate.

By this time Bill Tindall had retired, Lynn Dunseith, I'm quite sure, had retired, yes, and Ron [Ronald L.] Berry was up as the director. He had been the division chief in MPAD, which is still around, Mission Planning and Analysis Division. Anyway, I ended up being deputy director for a while, and [S.] Milo Keathley took over Building 12, or Data Processing Systems Division. Elric was still head of the Spacecraft Software Division and John Aaron was still no longer in that directorate, he was running the Station Program.

Okay. So after Challenger, John came to talk to me and said, "We're going to be moving the program office to Washington," and would I consider going up there for a year to help move it. Looking back now, I haven't the slightest idea why I agreed to do that. It was an interesting part of my life, most entertaining, most challenging, but it did.

Our two daughters were in either grade school or intermediate school, but pre-high school, and we weren't going to move them. Sue, my wife, had gone back to work by this time. She was working for Hernandez Engineering, as I recall, which is still around. She decided to go back to school full time, to quit work, go back to school full time. She wanted to get a CPA, accounting degree and a CPA [certified public accountant].

So the timing was about right. The kids are old enough to kind of take care of themselves, but they're not old enough where we're going through the driving and all that kind of business. So we agreed to do that, and I'd go up there and come home from time to time and she'd go back to school full time. She ended up doing it as a NASA co-op and ended up joining NASA. We'll talk that later a little bit.

But I went up to Washington for what I thought would be a year. It turned out to be almost two years. They had selected a fellow named Andy Stofand [phonetic], who had been the center director at Lewis Research Center, which is now Glenn Research Center, to be the new Level A. Had to change the name, so they called it Level One. And a fellow named Tom [Thomas L.] Moser to be the new Level Two, which had been in Johnson but now was going to be in Washington. In other words, the program director. Tom had been head of Engineering Directorate. He had been deputy or something while I was there, so he knew me. Remember, I was the Romeo and Juliet guy. He knew me. He was most pleased to grab me for a while. He had by this time moved up to Washington permanently himself and he was deputy head of the federal program, and when they did this reorg they grabbed him out to run the Station Program.

Well, this was a real mess. The NASA Headquarters welcomed us with open arms and no help. I'm exaggerating. There were a lot of well-intentioned people that tried to help, but it was not real smooth. When you make a movement like that, that's kind of a knee-jerk reaction—this was in the last few months of 1986, as I say, right after the Challenger in January, the report in the spring/summer, and the action in the fall.

We ended up in offices all over the area. There's a big shopping center at L'Enfant Plaza, one of the metro stops right across the street from the old NASA Building, and I remember a bunch of folks were there. I ended up in the Department of Transportation Building, which is a block the other direction, also at the L'Enfant metro stop.

In fact, the metro stations in Washington can be huge, and many of them have four exits going in four directions. So when you hit the street, you can be blocks away from another entrance to the same station. I remember clearly, on very cold winter days, people would actually pay the dollar just to walk through the station because you could walk out of the NASA Building across the street, down the escalator, and then you'd be walking two blocks underground and come out at L'Enfant Plaza inside and stay warm. Sometimes it was worth a buck to do that. And that's not a bad picture to paint, because it was kind of like that. We were scattered around these different buildings, the idea being that we were going to move to a permanent location, but we had to go through a competition of some kind.

It ended up being Reston, Virginia, which is, I don't know, about fifteen, twenty miles outside of downtown. It's on the way to Dulles Airport, just about five miles before Dulles Airport. I ended up living in a hotel for six months. I couldn't sign a lease anywhere, didn't know where I was going to be. And had a lot of interesting times working our way into the infrastructure of NASA Headquarters, particularly for a guy who, like me, was clearly temporary. I was not going to be there but for a year.

What we did was to pick up—I picked up three major functions there. One was this SSE, which by this time was a contract with Lockheed. In fact, Dick Parten won it. He had retired from NASA, gone to Lockheed, and led the bid team for Lockheed, and won it, against IBM, I might add. They were one of the big competitors. It was sort of interesting. Bear in mind that IBM built the onboard software and Dick Parten was the division chief that was the customer. No question that those two groups were well qualified on building flight software and support software environment. I'm sure that's what it stands for now.

The other thing that came along was the notion of something they called TMIS, Technical and Management Information System. The idea here was to have an infrastructure of everything from e-mail to simulations and so on for the Station Program. Now, again, there was no prime contractor, so the idea was to furnish computer services, software engineering environment, and what have you, to the various elements of the Station Program that would be building it.

This was a great idea, but not practical. In the long run, I think it just didn't work at all, the reason being—in fact, years later I called it the TMIS effect, even though it was an effect that happened for SSE and everybody. When you take from the big Marshall Space Flight Center [MSFC, Huntsville, Alabama] Space Station Program and the Johnson Space Center piece of the Space Station Program and the Kennedy [Space Center, Cape Canaveral, Florida] and the Ames [Research Center, Moffett Field, California] and all this, when you take from them and from what would otherwise be embedded in their budget all of the computing services and lump it together under two big programs, one called TMIS and one called SSE, you create a horrendously big computer budget that has no product. It's not seen. And of course, that money is summarily removed. What would have been in the budgets for the centers was taken out of the centers' budget, because we had a GFE, you know, government-furnished equipment. "We're going to give them these services." Yeah, right. Well, I mean, the intentions were quite good, but it created a target and it created a target in a post-Challenger era, a Space Station startup, a new administrator, a lot of new managers.

I remember one of the first things we went through were huge budget cuts and a lot of reviews. In fact, interesting story. One of the big reviews was on TMIS itself. Why centralize all of this infrastructure computing for Station? The experiences had not been good in industry, had not been good anywhere to speak of in doing this.

The notion of outsourcing didn't come along until about ten years later. The new senior NASA management, which was Jim [Dr. James C.] Fletcher, who was a returning administrator, and Dale [D.] Meyers, who had come out of Rockwell, I think, as a senior exec, they were both quite concerned, and we created this huge budget.

They decided to do an independent review and I said, "Fine." We were out at Reston by this time, which happened in the summer of '87. I said, "Pick your committee. We'll be ready." Boeing Computer Services later sold to SAIC [Science Applications International Corp.]. Boeing Computer Services, which was the prime contractor. A fellow named Peter Dooby [phonetic] was the head.

I thought we had a good system and a good story. We did. You know, when you're challenged about whether you should exist or not, sometimes that's the best thing that can happen to you because you have to look at your belly button and answer some very serious questions about why you're doing what you're doing and why you're spending what you're spending. So it was a very healthy review.

But the funny story in here was that they, of course, were not going to let me choose who was on the committee, and so they went through this search for who should head the committee and who should be on it and who were the real experts and this kind of stuff. They found a guy that had worked for NASA in the past that was there now working in Washington, D.C., with the FAA. I don't remember the name of the company, a big company he was working for. I have a name block on the company, but the fellow's name was Bill Tindall.

So they asked me if I thought Bill Tindall would be a good person to be the chair of the committee and I said—you know, how do you keep a straight face? I did. I said, "Well, yeah, he sure knows a lot."

"Do you know him?"

"Oh, yes. I know him. I knew him back at the Johnson Space Center."

So they made Bill the head of the committee. And of course, Bill chose to be on the committee with him people that I hardly knew at all. I'm joking. I mean, I knew all of them. Now, lest anyone think that as a result the committee was wired, it was not. Sometimes that's the hardest committee because you know each other, and Bill was very, very clear about, "The fact that we know each other may make this a tougher review." But the point was, we had nothing to hide. It was a good review, and what made it really work was that there was trust because most everybody knew everybody. I'm not going to lie, not going to spin things badly.

Anyway, the TMIS survived that review, but the budget did not. It was cut significantly, and as a result—and SSE was the same way—much of what had been promised to all the sites was not given to them as the years wore on.

Now remember, I was there for a transition period. It was the summer of '88 that I was gone. I came back to Houston. It was good. I mean, the Reston program office existed, things were rolling, but you could see, it was easy to see even then that there wasn't enough money to fulfill the expectations of all the folks at the sites.

The TMIS effect, as I still call it, was that not only did centralizing it for both TMIS and SSE create this huge budget that was always a target to cut, but when you say, "I'm going to give you everything you need to do your job relative to computing to the sites," they immediately assume that they're going to get everything they need, even if they forgot to tell you they needed it. So you get this diverging effect. The needs of the people, their expectations lies, and because of the budget centralization, what you're going to provide falls, so there's a gap that grows. Even if it was perfect at the start, a gap grows. "Well, we don't need to add a budget for that. Heck, TMIS is going to provide it and if they don't, we know who to yell at. Not our fault."

I think over time that became a very, very serious problem. And in fact, in '92, I think that's when [NASA Administrator Daniel S.] Goldin came in, and they completely redid the Station Program, buried what was called the Freedom Program, another metamorphosis, the TMIS and SSE contracts were both killed. They were both finally put to sleep, and major reductions in the computer stuff happened.

That was a most interesting experience for me. It was one of those where I didn't get a chance to follow through. I helped start something and then had to move on, go somewhere else. In my case, it was coming back home. The oldest daughter was turning sixteen. You want both parents home when the daughter is turning sixteen, trust me. That's when they drive. That's when they start driving.

That was in '88 and I was back home. When I came back, Elric had moved up to be deputy director. I had been in the Senior Executive Service [SES] for four or five years by this time. I got that before I went up to Washington. I concluded that I didn't mind not being a supervisor. "Hey, Ron, why don't you make me associate director for something or other and I'll go plan future things?" And they allowed as how that was a good idea. I was kind of a free resource coming back.

Over time, as I recall, we did some reorgs, and Elric was grabbed to go move into one of the program offices, Shuttle, as I recall, and I did become deputy director under Ron. I don't remember the exact time frame, but early nineties, within two or three years after I came back.

By this time, Sue had joined NASA. She'd gotten her degree, accounting degree, passed the CPA first try with flying colors, which was unheard of. But she was in her, must have been her early forties and accounting companies don't like people that know what they're doing, that have opinions. They want them young and green so they can work them to death and train them like the military.

So she'd been in the co-op program with NASA at the school, so she joined NASA. Very, very low pay. They gave her no credit for her prior experience. Government does that sometimes. But she rose quickly and she was working for a woman named Dee [Deidre A.] Lee, who was head of a division here in procurement.

Dee later moved to Washington, became head of NASA procurement, a few years ago moved over to OMB [Office of Management and Budget], became head of the Office of Procurement Policy and then just a year or so ago moved over to DoD as head of procurement there. I mention Dee's name because Sue was asked by Dee if she would go up to Washington for a while.

Now, I want you to picture this. I mean, Sue kind of says, "My turn." And by this time, this was in early '92, I think, the daughters were both driving. One was in college, the other was in high school. I mean, it's all over by this time. Mama's not going to be missed that much and Dad can certainly clean up their messes. No problem.

So we allowed as how that would work, and she ended up going up. She thought she was going to go work for procurement, but she didn't. Dan Goldin had come in about this time. George [W. S.] Abbey had come back from a White House tour, as I recall, by this time, to help bring Dan into the agency. Aaron Cohen, who had been the center director, was moved up to be deputy administrator, also to help in this transition. This was at the end of the George [H. W.] Bush, Sr., administration. They grabbed Dee out of procurement. She wasn't head of procurement at that time. This was early. She had just moved up to Washington from JSC. They moved Dee over to be on Goldin's staff. So we thought Sue's move was off because she was a procurement person, but I remember clearly Dee calling her up and saying, "Well, are you still going to show up next Monday?" and Sue going, "What?"

I'll make this brief, but she ended up going up there as a—by this time, she was maybe a GS-11 or GS-12, and after about nine months there—was it nine months? No, just six months. After six months, they asked her if she'd stay an extra six months and do a full year as Dan Goldin's executive officer. Okay. I mean, how do you pass that up?

She did, and she was about the best he'd ever had in that regard. Usually people that play the executive officer role are very senior people that want to move up, you know, maybe GS-15s that want to be in SES or something, so they're a little bit picky about what they do. She was not. The way I like to say is she's the only person that I've ever seen that didn't mind getting somebody like Goldin a cup of coffee or writing his speech. And writing speeches is tough. You've got to understand the policy and know what's going on, and she could work from one end to the other with no problem. So that worked very, very well. She was very successful in that.

During this time—I can't remember the exact parallelism, but it was during this time that I became deputy director under Ron Berry. We had a lot of, what I remember, big budget wars.

NASA was really going through some hard budget times, and I was focusing on the institutional side again, only now we were at the directorate level.

In this time frame, the agency started looking much more closely at its computer investments, and as I dive into that, this is the notion of senior IRM [Information Resource Management] official, and later becoming CIO [Chief Information Officer] kind of thing.

I think it's a great time for us to take a break.

RUSNAK: Okay. [Tape recorder turned off.]

GARMAN: I have to remember where we were, since I sort of cleared my mind a minute ago.

RUSNAK: That's all right. You had just brought up the topic of the chief IRM officer and getting into the Chief Information Officer, that kind of thing.

GARMAN: Ron Berry, as director, had the title or the function of senior installation Information Resource Manager. I think that was what it stands for. In those days, information resources management, or IRM, was the buzzword for how you manage information technology.

By this time in the history of computers, it had become clear that information was a real asset. Most people didn't think of that. I mean, you think of file cabinets full of patent rights and drawings. You knew that was valuable, but with computers, when there's something you can't touch, it's just bits and bytes on a machine, the abstract definition of information became a lot more real, as companies would go bankrupt because their computer crashed and they lost all their data, and there was this big thing to be paperless.

I have to mention something on paperless. Those of us in the business have laughed at paperless for years. The number of devices at the Johnson Space Center that could print ink on a piece of paper were small. It was typewriters in the sixties and the seventies. Not many people had them, generally, secretaries. I was one of the few engineers at the center—I don't know if I said that story. When I came to the center in '66, I knew how to type. I was the only engineer that knew how to type, and I insisted on getting a typewriter. There have been a lot of battles to get rid of sexism in the industry but I want to tell you, in the mid-sixties it was alive and well, and engineers don't get typewriters. The girls get the typewriters. And they just wouldn't give me one, and I remember just arguing vigorously that this was crazy. "I'm not going to type the final stuff. I'm just going to type. I promise, instead of handing my handwritten notes, that they can't read, to the secretary to type up, I will hand them rough typing, with all sorts of errors, on a piece of paper. It's just that I can type faster than I can write. Please, can I have a typewriter?" They finally gave me one, and I swear, I was the only non-secretary, I was the only professional that had a typewriter, that I know of.

Anyway, there were a lot of—in fact, there's some stories about that, because I always had a word processor or something in my office, my whole career. It used to drive some of the secretaries crazy, because with a typewriter you can hear it go, and I type pretty fast, so there'd be a race, you know, how fast was my typing compared to—of course, I made mistakes, so it wasn't fair. Not a fair contest at all.

Anyway, the number of printing devices, the devices that could print at the center, was fairly small, for years, until we got computers and we went paperless. There's an anachronism there somewhere. The fact is that paperless really meant that the primary storage was not necessarily on paper, you know, what you went to to do your work. But everybody ended up having a printer on their desk or down the hall, whereas in the earlier days, the printed form was the final form and handwritten notes were the drafts. Nowadays, the final form is on a computer and everything you print is the draft, right? You type it, you print it. Unless you're really good at seeing everything on a screen, but most people use printers to print drafts. That was quite a change.

Anyway, it had become clear in the industry that information was an asset or a resource, and so to raise the level of scope, not just supplying computer services and networks, but let's think about information technology and its real purpose, which is managing information resources, information resource management, and one of the many continuing quality-improvement programs that the federal government honestly tries to keep doing. IRM became a big thing. I think it actually started in the military side, but became a big deal.

Another effect driving that change of view was actually out of the software engineering world. I remember clearly in the late eighties—I think I was still in Reston—that the Department of Defense published a paper that plotted—the operative part of this paper was the gap or lack of skills coming out of schools, on computers, and they were focusing on programming. They had a plot that showed the rising need, and it was a line that was rising. I don't know if it was straight or exponential, but it was rising rapidly. And then beneath it, of course, was another line plotted by year that showed the number of qualified programmers or skills coming out of school, and the gap was growing.

Well, I enjoyed those kind of plots because I took that, and of course, this was in '88 or '89, they had plotted it for five or six years. I took it and plotted it out, same curves, out to the year 2050 or something, and the gap by that time, if it continued, showed that every man, woman, and child would have to be a programmer, and I used to joke about that. I thought that was really funny.

And then one day I was thinking about that and I concluded it was absolutely right. We all are. What was happening in the industry was that more and more, instead of using computers and the computers did all the work and all the programs were all prepared to do the work for you, what was happening is that computers were becoming interactive. Anybody that's ever taken a spreadsheet and put an equation in and adds up numbers in columns and rows has written a program. That's programming. In the early days of computing, you didn't do that. The program did it for you or it didn't do it at all. What was happening was that the level of interface that people have with computers was getting higher and higher and higher, as it is today.

So information resource management had another element to it, and that was, where were we headed in terms of what people did, what did programmers do, what did the software industry provide, where were computers going. Boy, that was fun. I thought that was a really neat thing to try to keep my eye on, so I volunteered to assume that title and really dive into it.

Every federal installation had to have a senior IRM manager. You had to name one, point of contact. And certain reports were required to help the government understand where it was in the business of information resource management. That kind of function at the Johnson Space Center fell to this directorate. But we had to report on all computing. It didn't matter if it was in our directorate or wherever it was. I realized that the idea of trying to get some function that looked over all the computing was a good thing.

Let me explain myself. In most corporations, in most government entities anywhere, computing tends to divide itself between the institutional or infrastructure kind of computing and mission computing, and often there's a third called engineering, and they never talk to each other. They don't. The engineers refuse to work with the institutional folks. The mission folks—now, you think by "mission" I mean like mission control or something, and I do at the Johnson Space Center, but what I really mean is the mission of the organization. If you think of an oil refinery, for example, the mission computing is the computers that actually run the refinery process, and they're totally distinct from the computers in the office and they're also often totally distinct from the computers in the engineering organization that's working on a new refinery or developing new techniques

This is a silo effect, the term "silo" meaning insulated, don't talk to each other, replication of effort, expensive, not efficient. As I've mentioned, I'd seen a lot of that at our center and across the agency, often because of the industry. DEC computers won't talk to IBM computers and PCs won't talk to Macs and all that kind of stuff. So I thought it was a great idea to grab hold of a function that had to do some reporting over all computing because that gives you some leverage to try to get people to talk to each other.

About this time, the term "CIO" was beginning to appear. Every time there's a new function that everybody thinks should happen to improve quality, they'll take the word "chief" and the word "officer" and stick a letter between it. You know, chief executive officer, chief operating officer, CEO, chief financial officer, CFO, and they came up with information, CIO. Now they have CTOs, chief technology officers. They even have CKOs, as we move into the notion of knowledge management.

But the idea of a CIO in industry was to elevate information resource management to be a very senior position and to do the TMIS effect, which I thought was stupid. Been there, done that. That is, take all the computing in an organization, put it under one structure, have it provide computing to everyone to avoid replication, to get rid of the silos. That's a way to do it. Hand it all to Joe, or Tom, or Sally. Have it all in one spot. I'd been there. I'd seen what happened. You get this big budget, it gets cut, the expectation gap rises, people don't get what they need, and it doesn't work. So my conclusion was that it would be a far, far better thing if the CIO function were an oversight function, like an audit or a review. Had some clout, maybe, but not accountable for providing services.

Well, it was getting near the end of Sue's term. This was the summer of '93. She came home in October. I was a cheap trip. She had an apartment. Okay? I didn't mind going to Washington on a trip, and it was cheap for the government because I didn't have to spend a hotel. NASA was trying to figure out where to go with IRM, and I volunteered to head a committee to take a real look at what the agency should do. So during that summer I spent a lot of time in Washington, D.C., chairing a committee that had some notable members on it, John Lynn [phonetic] from Marshall and Joe Bredecamp [phonetic], who was at Headquarters but had come out of Goddard. Many other notables in NASA in the information technology arena. Some of these folks had some mission experience, some were engineering, a lot of them were institutional computing infrastructure. By institutional infrastructure, I mean things like the accounting and the word processing and the PCs. They hadn't really merged yet as they have today. They were still pretty distinct.

And we concluded that NASA, as expected, NASA had a lot of replication, unnecessary. For some strange reason, the recommendation of this report was that NASA really should create a CIO and that each center should have a CIO, and that in the agency and at the centers, the CIO should not own all the computers—TMIS effect—but should have the oversight responsibility, and it sold. It sold.

The next problem was trying to talk someone into being CIO for the agency. I was not interested, of course. I'd already been up at Headquarters for a while. Sue was up there and coming home. So we talked John Lynn into it and he became the first CIO of the agency.

Back at the center, Sue's coming home, Carolyn [L.] Huntoon is about to become center director, George Abbey is about to become deputy center director. Sue knows all these people, so they end up making her Carolyn's executive officer, as she was Goldin's. And of course, she knew NASA inside and out, so that didn't hurt at all, didn't hurt Carolyn Huntoon or George Abbey at all.

We went through a reorganization at the center and formed a CIO office and I got to pick twelve good people. That was the number I said that could do the job, because remember, no contracts, no services. This was an interesting experience for me. I had been a second- or third-level supervisor now for twenty-some years. All of a sudden, I was a first-level supervisor again. I had no managers working for me. I had senior people working for me. I had an ex-division chief. The Center was also going through a restructuring to lower the manager-to-supervisor-to worker ratio, so we had a number of folks that were no longer supervisors. They didn't lose any money. They did a flattening of the organization. That was another of the quality efforts.

So we formed this CIO office. Of course, I'm sitting here in recent history. I'm saying it worked very well. It's still working. But the idea was that there was a separation of power here. The CIO office had no contracts, ergo there was no contractor or empire to defend. As the former Governor of Texas used to say, "No dog in any fight." The former former. I'm talking about Ann Richardson. She used to say that. There was a separate directorate. It's still there, this Informational Systems Directorate that does the institutional computing, and other directorates that still do their thing in major computing. Mission Operations does the control center, and the Shuttle Program with the big USA [United Space Alliance] contract does the onboard computing.

These are all huge monolithic kind of things and they never would talk to each other, except that what we did pretty much across the agency was to empower the CIOs to set standards, and their enforcement power was they had to concur on all acquisitions. They didn't tell you what to buy, but they could tell you you couldn't buy what you wanted to buy. Okay, so authorization. That's fairly powerful. But there's no money involved. In the government, generally authority comes with control of budget or money. That's true in any world. But in a bureaucracy, if you have to get the approval of someone, that's also power, too, and it's a good way, in the government, of doing oversight kind of controls. You have to be sort of a benevolent dictator. You have to be careful.

In some cases, it has not worked at some centers because people were not forgiving. You can't pay attention to details; you have to look at the whole. At other centers, particularly non-Code M [Manned Space Flight] centers, by and large, they left the CIO function tied to the providing of services and it's usually institutional services. If you're head of the institutional organization, the mission folks, when you try to tell them they can't buy something, they say, "Stick to your own turf." You just don't get the authority.

At this center, they kept it separate, they have kept it separate, and it's worked fairly well. But it also led to the rather famous PC-Mac wars. I ended up being the cause of—I say that loosely—not personally the cause, but the policy ended up being the cause of an incredible e-mail storm to all the congressmen and all of the NASA management.

We had concluded that it cost a lot of money to have many, many different varieties of computers trying to do the same thing. So I put in place a policy that said we wanted to find a single solution wherever possible for every function, and that what we'd end up doing is pick what was the most popular, not because a popularity contest is the right way of doing it, but because in the computer industry in the mid-nineties, if the company was still alive, it probably had the best products, and if it was selling a lot.

This was like, you know, everybody bought IBM in the seventies. They may have been a little more expensive, but they provided the best products and they had good service. The Johnson Space Center was like 75 percent PCs, maybe 20 percent Macintosh, and the rest were UNIX and other things. So I said, "All right, for desktop computing, as computers get old and go away, we'll replace them with what are now called Windows PCs." I did not realize the religious significance of Macintoshes in some people's minds at the time. This was right about the time that Windows 95 came out. We were privy to that, had been using it for a year before it came out, realized that the major advantage of the Mac world, which was the very, very easy user interface, that they'd gone static, that the Microsoft folks had sort of passed them by or were passing them by. So we had basically an equivalency in user interface, ease of use, and the huge majority in infrastructure. I think it was 75 percent of the machines. So it seemed like a no-brainer to me at the time. In a business sense, it was a no-brainer.

Apple Computer was in trouble at that time. They actually had a senior official that was head of evangelism. That's actually what he did. It was his business, evangelizing the Macintosh computer. He and they ran an operation where they let people who felt that the Macintosh was being unfairly moved out or what have you, they set up networks, using e-mail, which is very popular, and websites and all that, to pass the word.

Hence, the e-mails that got sent out became, "The Johnson Space Center is having trouble. They're losing their Macs. There's some guy named Garman that's trying to get rid of them all. Why don't you write to your congressman and let him know how NASA is all screwed up." And of course, by the time this goes to the fourth person, the exaggerations and the embellishments are tremendous.

We had the Inspector General all over us, we had congressmen all over us, and stir aground. I remember it got scary once. I got a bomb threat and we had to get security involved. At NASA, all the parking is outdoors. There's really no security. Everybody knows where you park if you're senior, because you have a reserved spot.

At this time, Sue is number three at the center. She's associate director, under George, and I'm CIO, one of the few cases I know of where we had a married couple both as direct reports to the center director. So when one of us gets a bomb threat, it's pretty scary. I don't know that it was real. You never do. But it ceases being a game of philosophy or politics when it starts getting physical.

Fortunately, by that time, things had toned down a little bit. We started doing what diplomats do. You start negotiating, compromising, not giving up, but you start making steps to salve the wound, so to speak. Really, all we did was to write down policy, which said that all of the exaggerations were not policy. If truth is at six and all the statements are being made at ten, if your compromise is to go back to six, you haven't lost anything, right? So we went back to maybe five and a half. I'm making that scale up but just to make the point. It was most entertaining to have to write down, to agree to write down, that Macintoshes were allowed at the Johnson Space Center; in fact, they were promoted.

If you're going to have a single solution for a function, it is still one of the best computing devices to use for editing video. If you ran a video-editing operation at the center, for Christ's sake, please buy a Mac. No problem. If you were doing office automation kind of things, please buy a PC. If you were doing engineering work, please buy a UNIX machine. That was the notion. So it was entertaining to walk through all that.

Backing off of the more personal experience in that, the industry was really going through a change and all that happened to us is that—the government usually doesn't get ahead of industry. We got a little bit ahead. We got a little bit ahead. Today if you walk through a computer store you're lucky if you can find Macintosh software in the store, and if you can, it's usually off in the corner. I don't mean that Mac is dead at all. What I mean is that the notion of computing here is that if you want to keep up with technology, if you want to be able to get hold of the latest technology to help you do your job, what you want is a device that you can buy software cheaply, and a lot of it. You want a lot of choice in what you can get.

So that has turned out to be the right approach. It's not necessarily that a Windows machine is the best machine. Microsoft's in a lot of trouble for a lot of good reasons, but today almost every piece of software written, particularly if it's new, doing a new function, almost always first comes out on the Windows machine.

So it did something else. Both the Mac and the Windows did it. They did something that we had tried to do just for the Station Program and, early on, just for ADA. It's the notion of separating even further what programmers do to write applications, to separate them from having to know what machine they're running on.

An almost anecdotal example I like to give is Lotus 1-2-3. Used to be Lotus would only run on a completely compatible IBM PC, DOS, DOS-compatible machine. And in fact, the ability of a machine to run Lotus was often a measure of whether it was IBM-compatible or not, whether or not it ran on an IBM machine. It became the measure of compatibility, as often happens on a popular program.

Why do I bring Lotus up? Well, Lotus, in the mid-eighties, was delivered with many, many diskettes. Many of them were the printer drivers, because Lotus had to write a different driver for every single printer made, and they supplied it with Lotus 1-2-3. So you loaded the program and then you looked at what printers you were going to print to and you pulled the disk that had that driver and loaded that driver so your machine could print to it.

What Windows did, and the Mac, I think, did it first, but everybody was headed this direction, what it did was to elevate the level of the operating system. In that kind of environment, the applications programs do not need to write the drivers for all the printers. They don't even care. They print to Windows, and Windows prints to the printer.

So you know today, you don't even think about it today, if you buy a new printer, you install the printer to Windows, don't you? It's Windows that gets the printer driver, and everybody wins. That is, if HP [Hewlett-Packard] comes out with a new printer, they only really have to write two drivers nowadays, one for Windows and one for the Mac, that's it. And they're pretty common, by the way, under the coverage. The applications folks, if they write for both a PC and a Mac, they write it once, port it to the other, and they print to the operating system. They don't have to write all the drivers. It's a lot more than that. It's not just printing. It's also what's on the screen. They used to have to write every single piece of code for everything that appeared on the screen, and you don't do that now. The operating systems take care of that. You know, paint a square this big. That's why Windows was called Windows; it's squares on the screen.

So there was a tremendous movement in computer technology during this period that allowed this separation in the computing environment or platform from the applications programs. Boy, this was déjà vu all over again for someone like me, because that's exactly how mission software, which I'd been involved in, had been written in prior years. You know, everything specifically for the specific platform or simulator or wherever you're at. Very, very expensive.

And when you scale up, and you have 10,000 seats at the Johnson Space Center, you want it to be real cheap per seat. You want to get the cost down. So this was part of the CIO function, part of the notion of trying to push the technology, was to not just reduce the plethora of solutions that couldn't talk to each other, get some commonality and some interoperability, but also try to drive out a platform on which people could do kind of what Lotus did.

You buy a copy of Lotus, or today, Excel, and you probably use 1, 2, 3 percent of the functions in there. But what you know is, if you need to do something, it's probably there. The issue is to open up the book or go to the help screen nowadays and ask how to do it, and it'll tell you how to do it. Every once in a while, you may have to go out and buy another thirty-dollar package to add in.

That's incredible, compared to ten, fifteen years ago. It's absolutely incredible. You want to talk to your machine today, you can go buy a program for fifty or a hundred bucks, and you load it, and you can talk to a microphone and it'll run Lotus, it'll run Excel for you. The fact that these programs can glue together is amazing. It's absolutely phenomenal, and it all has to do with building environments like the Mac does and like Windows, in which the level of programming gets higher and higher and higher.

Well, that was the big challenge of the mid-nineties. As we got moving ahead a bit, the next one was that we were spending way too much resource on it. The industry was beginning to discover outsourcing, outsourcing being the notion that you take functions that you do that are not part of your core reason for being as a corporation or an industry or a government entity, and instead of having your own people do it, you take that function and say, "Let me go find another organization that does it as their mission. It's their core mission and I'll use them to do it."

One of the famous early examples of this was General Motors, which outsourced all their computing to EDS. That was the big [H.] Ross Perot company. Because General Motors had a lot of engineers and a lot of managers involved in IT, in information technology, and General Motors is not an information technology company, they build cars. EDS was a big computing company. What they did was provide computing services and had a lot of computing talent. So it was possible that someone who was in computing could rise to the top of EDS. There wasn't a chance in you-know-where that the person who's good in computing in General Motors would ever rise to be the head of General Motors. It's just not a career path and it's not part of the core function.

So General Motors decided to hand over all of its computers to EDS, and I mean, lock, stock, and barrel. People that worked for GM woke up one day and were working for EDS. They just moved them over. They came through and EDS owned all the General Motors computing equipment. They handed it to them because they wanted them to refresh it and make it new. Very shocking. It was not well done. It was well done for the time, but looking back, it was not well done. Very threatening. The GM folks said that they come in one day and here are these EDS folks coming through and ripping GM tags off of all the equipment and putting EDS tags on, saying, "It's ours now."

"Wait a minute, it's my computer. I need to use it." You know, you get that sense of loss.

Well, NASA was facing some of the same issues. It was really time. NASA had always contracted. The Johnson Space Center here has always had upwards of 10,000 people in the campus and never had more than 2,500 or 3,000 civil servants just on campus, much less outside the fence. It's always been a lot of contracting out.

But this is the notion of handing over the whole function. NASA had not done that. NASA does cost-plus contracting. People are paid by how many hours they work, by what it costs. Is there incentive to reduce cost there? No. If I'm a contractor performing a function, if I can figure out how to do it with half the cost, what happens to me? I get half the income. What's the incentive?

NASA puts award fees on it and that helps a lot. It does. But it's not the same as going fixed-price. If I agree to do something—in my case, computers, if I agree to take care of your desktop computer for a couple hundred dollars a month, that covers everything. We'll replace it every couple of years, all the latest software, help desk, repairs, everything, it's all covered. You don't even own the computer. The customer gets a very well understood budget and if the contractor can cut the costs for doing it, what happens? Straight profit, which incentivizes them to do that and then what happens on the next phase of the contract when they have to bid again? They lower the price.

So we caught onto that. If there was anyplace that fixed-unit price contracting could work, it would be in computing. We concluded that outsourcing was a really good thing for the Johnson Space Center and started a program to try to do that. This was in '97, I think.

The agency thought that was a good idea, too. If we could outsource together, all the centers together, not only could we reduce costs, but we could force everybody to use the same products, get rid of some of those silos. Well, that's very threatening to a center, particularly when there was no agreement whatsoever on what the products would be.

The long and the short of that was that the agency adopted one of its major institutional programs called ODIN, Outsourcing Desktop Initiative for NASA. Again, as CIO, I was not involved in the contracting at all. The concept was where I was, this Information Systems Directorate. But in putting that out, NASA attempted to create a single contract. But unlike the TMIS mistake, where we'd all be in one budget, each center did its own. It's kind of like you can order your own meal, but we're all going to go to the same restaurant, okay, as opposed to, we're all going to go to the same restaurant, and here's your meal. You get it; you don't get a choice.

It has worked reasonably well. The biggest problem has been a switch from cost to fixed-price contracting because a NASA person is very used to saying to a contractor, "While you're here, could you do this for me, too?" In a cost contract, of course, no problem. If the person takes an extra hour to fix something, no problem. And it's probably a worthwhile thing to do.

In a fixed-price contract, it's kind of like, while the plumber is there at your house, fixing the sink, you say, "By the way, I've got a leaky faucet over here. Would you mind fixing it, too? And don't charge me any more."

The plumber says, "I don't think so. You have this coupon for X dollars to fix the sink and that's what I'm here to do. If you want me to fix something else, I've got to charge you more."

You say, "I don't think so. I used to have a plumber that came here and did everything I wanted." No, it doesn't work that way anymore. It doesn't work that way anymore.

So that's been a rough row to hoe. Change is always tough. Change is always tough. That's been difficult. That's kind of where the center is, where the agency is, where the center is. There are several inter-center programs they're trying to do. Another one is called the financial management world, which is another big infrastructure program. That's had many false starts. I was involved in the one early successful one. They used mainframes, stable environment, back in the eighties. In the agency, they're still using many of the programs out of that. It was called AIM, as I recall. I don't remember what it stands for. Administration Information Management, how's that? Something like that. System. AIMS. There is another anecdote in here.

RUSNAK: If I can stop you for just a sec, so we can change out the tape. Then we can come back and finish it.

GARMAN: Yes, absolutely. And I'm finishing up, by the way. [Tape change.]

GARMAN: Okay. In '92 or so, it was that same time period, lots of change. Goldin coming in. Sue was up in Washington. There was an effort to reduce the costs of data centers. We still had a lot of huge mainframes across the agency, and no one could agree to do it. But among the Code M centers, the Marshall Space Center in Huntsville, Alabama, had really, really wanted to become the IT infrastructure center for the agency, make it a reason for being for that center. My own personal opinion, that the three big Code M centers, Johnson, Marshall, and Kennedy—Stennis [Space Center, Bay St. Louis, Mississippi] is the fourth—but the three big ones, it's always been hard to justify three. It's a lot easier to justify two.

So there's always the threat of closing down, and it's always Marshall that ends up being threatened to be closed. It's a smaller state, not as much political clout as Florida or Texas. You've got to have Kennedy to launch them, and you've got to have Johnson because of mission control. Johnson used to be, and still is, a big engineering center. They can build spacecraft. Kennedy can do some building and refurbishing, so what do you need Marshall for? We need Marshall. That's the rocket science. That's the Wernher von Braun world. But still, there's a threat there. There's a threat. Unless we're building something new, their role is diminished.

So there's a lot of effort to create a reason for being, and I think the administrative computing is one they've created and it's a good one. It's a good one for them to do. They picked up networking years earlier for the agency, wide-area networks, and so in the early nineties, they wanted to be the data center for the agency, the single data center. Well, you know, single data centers don't work. Tornadoes happen in Huntsville, Alabama, and you need some replication.

But in our case, at Johnson, we had just built a huge data center, Building 46, at the Johnson Space Center, and had filled it with all the mainframes and computer operations. That was part of what I was involved in just before I left and after I came back from Reston. Very expensive.

We looked at the trend in the industry. The only way you reduced cost in big data centers—data centers meaning the big raised-floor areas where you see all the big mainframes, "glass house" they sometimes call them because they always used to have glass walls so that the senior people could take their customers through and show them all this equipment, without letting them in the room. The only way you could reduce the cost of a data center was to grow it, because what was happening in the mainframe world was that in the old AX-plus-B equation that exists everywhere on anything, that if you have one of something, it costs B to get it. "A" is the factor for adding one more. The incremental cost of adding something is usually pretty small. It's the first one. The first step is the lulu. Having a kid. The first one is real expensive. The second one isn't as much. You know, incrementally. Not a good analog, but very much the case in mainframes.

Once you have a mainframe operation, you've paid a lot. To double the size of it does not double the cost at all, particularly because mainframes were getting bigger and bigger. By and large, you could throw out your old mainframe and get a bigger one, and the cost of it wasn't that much compared to the maintenance cost of the old one. The new technology meant they were easier to take care of, they didn't fail as often, so your overhead was down. The electronics were getting better and better so the power usage and the cooling requirements and all that went down.

So the only way to reduce cost in a data center was to grow it. But we weren't growing. The agency wasn't growing that way. The agency's computing resources were growing phenomenally, but not in mainframes. They were growing in servers and desktops and all that. The industry was changing.

So the way to cut costs on data centers was to consolidate. Take data centers that existed, move them together, that creates a bigger data center. That's the way to cut costs. I agreed with that. I had no dog in the fight. I remember when the agency, when Code M, at least, said, "Look, the agency won't do it. We're going to do it. We're going to consolidate all of our IBM-type mainframes together." And Marshall said, "We want to do it," and I said, "That's a great idea." And they thought I was loony, because we had one of the largest data center operations in the agency, here in Building 46 at the Johnson Space Center.

Bear in mind, I had been to Station and back. I'd watched computers come. I was not the CIO at this point, but I still was in the oversight kind of function. So I finally said, "Well, look at this way. I think it's time to put all the elephants in one graveyard, and if Marshall wants to run that graveyard, it's fine with me."

Now, there's a little spin on that, in that I'm trying to make it like it's old stuff and we don't want it. It's not. Mainframes are still around. They'll always be around. But I had to change the view of the mainframes at JSC from being part of JSC's reason for being, a big corporate asset that if we give it away, maybe we'll lose, to being something that's a drag. It's a drag. So why don't we go ahead and move it, let somebody else take it over.

It was one of my first encounters with George Abbey, too, I remember, by the way, and one of those, "Why did you do that?" kind of questions that he does. "Because it's a good idea." He said, "Oh, okay." I explained and that was that.

This was long before the PC-Mac wars, but this was another one of those, you make a decision to support something and it's, you know, path not taken. You never know if it's the right thing, but by God, picked them all up, put them in trucks, and moved them to the Marshall Space Flight Center, all the mainframes. There's not one left at JSC. There are some but they're not IBM mainframes. There are some smaller machines, engineering. Building 46 is largely a server farm now, full of servers.

Anyway, that was another one of those interesting changes in computing at the Johnson Space Center. I think it actually put the center ahead because it's another form of outsourcing, right? We didn't eliminate people. So what we ended up with was a whole bunch of folks that knew a lot about running mainframes that didn't have a job. They were forced to either move on or learn about servers, desktops, networks, and that's where it's at today.

That's what you do when you're outsourcing. What you are going to hold on to, you want to focus on what you're really going to use, what you really need, and get your skills moved up to that. In the same sense that the only way to save money on data centers is to grow them, the only place that it's worthwhile to have that skill is in the place where it's growing.

So mainframe skills are still a good commodity at the Marshall Space Flight Center and unless that's a growing thing, let's put them all there. It helps everyone. It helps everyone. But these are hard concepts to get across to people. Change is always difficult, particularly if you're the one being told to change. It's like standards. Standards are fine as long as they're mine. Another one of those poetic things.

Anyway, as I'm trying to reach some sort of conclusion here, I think the notion of computing at the center has continued to, I think, to try to keep up with the industry rather well, simply because there's this continuous pressure to cut costs, budget, and there's enough people around that are always wanting the latest technology just because they want it. The fact that it makes them do their job better is beside the point, but they want it so they're willing to experiment. We keep having nuts like me that like change, so we keep pushing it.

I think I should mention one other major change in computing, and then we can wrap up with questions or whatever you want. It's security. Computer security is a whole new game. In the early days, and I mean early from today's perspective, not from my experience, in the eighties, because the computers didn't talk to each other, because IBM networks couldn't talk to DEC networks and couldn't talk to Xerox networks, because of all the silos, because we had to string wires individually, computer security was not a problem. I mean, what do you hack? What do you go after? There was no common highway. It's like if you're going to pick locks, wouldn't it be wonderful if all the doors in the world had the same lock on them? They do today. It's called the Internet. They didn't fifteen years ago. The Internet was something that was buried, used by DoD and universities, and it was all UNIX. UNIX today is still 95 percent of the hacking incidents in computer security because it's been around for such a long time.

So what happened is, as we became more and more interoperable, as the use of computers themselves changed—today, even at home, you don't think of buying a computer without a modem that you can dial the Internet on, right? Well, come on, five years ago, that wasn't the case. Just five years ago, they didn't come with modems. You had to go buy one, to add them in. They didn't come with CDs, they didn't come with sound. At least the Windows PCs didn't.

So what's happened is that nowadays a computer is viewed as being connected to databases and servers and Internet and so on, and interconnection means there's a highway, there's a path. If you can go out, you can come back. You can come back. If you can get to a server down the street for your data, so can someone else, and maybe it's not a person that should.

So the incidents of computer crime and computer hacking started growing enormously, exponentially, around this same time period, mid-nineties. It's been coincident with the Internet growth, just about parallel, as has been the availability of information.

Well, the CIO office, which had the oversight function, had no dog in any fight, ended up being the office that took care of computer security, and suddenly it became on operational function. It's a very difficult function.

The other kind of computer security was computer misuse. As more and more information became available on the Internet, it became more and more tempting to get drugged by the Internet. That is, to spend your time browsing the net when you were supposed to be working. Now, in the old days, when someone brought a newspaper in, or magazine, and was reading or sleeping at the desk, it was very visible, very visible. So they write rules. They say, "During your lunch hour, if you want to eat at your desk you can have a newspaper there and do it, but otherwise, we don't even want to see it. It doesn't matter to us whether or not you're actually going to read the newspaper. I don't want to see it, because there's an appearance that you are. It doesn't help NASA, it doesn't help us, it doesn't help you as an employee. It's just not good, so just don't do it." Hell, like dress codes. You can't push it too hard, but, for God sakes, you're not going to come to work naked, we're not going to allow that. There are extremes at which you just don't do it.

Well, on one hand then, we've ended up with more and more people trying to break into computers—that's the hacking side—and on the other end, we have more and more people using computers incorrectly, the worst, of course, being pornography. People just love to download pornography, especially if they're young guys. And some of it is not only despicable, but it's criminal. I mean, felonious. You know, felony level.

In the government, when an employee commits a crime, the connection to criminal law is the Inspector General, who calls in the FBI and all that, for investigation. So it got to be real entertaining, because what happened is that many, many people discovered they could get at all sorts of information on the Internet that had never before been available. I mean, for heaven sakes, it's one thing to be reading *Newsweek* at your desk and something else to be reading *Playboy*. And even twenty years ago, if you ended up taking a *Playboy* centerfold out of your desk and hanging it on the wall, that was viewed as sexual harassment in any time. Okay, you just don't do that, right?

On your computer screen, isn't that the same? Sure it is. Worse, in many ways. So whether criminal misuse or what have you, and not just that. People doing their stock market work, all that kind of business. What people didn't realize was that the very same devices that were put up to protect centers, organizations from hackers, called "firewalls," are devices that track use. The notion of a firewall isn't to block. The only safe thing you can do is to block everything, and if you block everything, the computer's not useful. It's like I like to say, the only safe computer is a computer that's turned off. The only safe driver is someone that's not driving. Life is a matter of risk. You take the risk to the level and the idea is to take the risk at the level that gives you the most return with the most safety.

So what firewalls do is they block the obvious stuff but they log everything else, because the notion is that if something does get through, you want to know where it came from so you can go back and get it. Stop it, fix it, take corrective action downstream. No different than taking good notes when an accident happens so you can go back and see exactly what happened and figure out how to fix it.

When firewalls log everything, they don't log content necessarily, but if you send an e-mail it tracks who it went to and where it came from, not the content, which is like the "To/From" address on an envelope. If you're going to a website, it tracks the website you went to and who you are. So firewall logs are perused by hand, automatically, all the time, looking for patterns of attack.

What our folks started finding were other patterns of attack, like particular individuals being at websites that were obviously not business. Most of the bad websites have bad names. You know exactly what they are and you look and you get this long list in a log of a particular desktop accessing strange websites. Whether they're pornography or non business-related didn't really matter. It's like Big Brother. You have no choice. When you see it, you've got to go do something about it. This isn't the "don't ask, don't tell" world. When you see it, you see it. So part of the big business of the CIO world was to try to convince people that it was not only wrong, but they were going to get caught.

I remember every incoming astronaut class, they would invite the senior folks to come give them a talk about learning the center. Pretty soon, my talk was all about security, because if there's any group of people that have to live by a set of standards that's a lot higher than the rest, it's the astronauts. They are the best of the best, and on top of that, they live in a glass house. Everything they do is seen.

I would give them a talk that went something like, "Don't, because you'll get caught. But if you're gonna anyway, here's how we're going to catch you." In other words, don't get caught. And telling them exactly how everybody watched and could see everything that was going on.

Most people don't realize that for efficiency, when you browse a website, almost all the material you look at is cached in your own machine. Cache means a copy of it is kept. You can control the size of the cache. You can cause the cache to be erased, if you want. Most people don't. What it does is, if you go back to a page, it comes up instantly because it kept a copy.

Another of my anecdotes is, I remember, in developing these little talks about if you're going to get caught, here's how we'll catch you, so don't, which is a good deterrent. I remember talking to the legal office. We were trying to make sure we weren't doing things that were improper, in threatening people or what have you, in explaining this. I walked over to one of the lawyers' desks and I said, "Here, let me show you," and I clicked on her machine and pulled up a list of every site she'd been to for the last month, many of which were not necessarily related to her job. She turned five colors of red, and I said, "Don't worry about it." Nobody's concerned about diminimus use of machines. There's nothing wrong with it, as long as it's not what you're doing instead of your job. There's nothing wrong with it, no more than there's anything wrong with pausing by the newsstand on your way to lunch or picking up the phone and talking to a friend from time to time. It's not that kind of environment. But you will get caught. I immediately ended up giving all the lawyers a tutorial on how to clean up their caches over time.

But back to the core of this conversation. The other major change in the nature of computing has been protecting computers and trying to get people to realize that while they're called personal computers, they're not. They're no different than the typewriter I had to fight for in 1967 as a young engineer. It was given to me to help me do my job, not to play. And computers are the same way. Particularly in this environment, where everybody has them at home, they use them, it can sometimes be artificially difficult to keep your personal data not on a government machine if you have a laptop or something.

So we spend a lot of time writing policies that kind of walk the line. Let's not be stupid, let's let people use personal stuff where it's appropriate to. On the other hand, let's not cross that line on appearances. And you can imagine, there's some very delicate and good stories I'm not going to tell that get into this stuff, about really innocent people that made mistakes and you have to figure out a way to get them off the hook, and really awful people that didn't quite make a mistake, so you've got to try to figure out, can we hook 'em? We'll wait, they'll make a big mistake some day.

Very much like law enforcement. Very much like law enforcement. It's a tough business, tough business, because people can really hurt their careers and you really want to try to keep them from doing it. The very thing that enables us to do so much more, computers, can do us so much damage. It's sad.

Well, with that, why don't we conclude, or if you have—

RUSNAK: Just to continue a little bit along these lines, why don't you tell us about leaving NASA, first of all, why you chose to retire and what you've been doing since. We can just start with that.

GARMAN: A couple of reasons. First of all, I'd never held the same job for more than two years. That can be viewed as a problem or not, I don't know. If you're hearing my voice you can't see the strange look on my face. I have always liked change and moving on, and to be challenged, keep doing new stuff.

The CIO office was formed in '94, formally. By the year 2000 I'd been in the job for six years, longest job I'd ever had, basically the same. A lot of change, you know, with computer security and things I was just talking about. Some successes, some failures, but by then I was feeling pretty good about it. Growing new people into the jobs.

I couldn't be promoted any further, no more money. Those are separate topics. That is, I was already SES level. No more promotion. The government has a cap on salary. My salary was already above a cap and so I was limited. My salary was larger than what I was permitted to be paid by law, so I was cut off. In the government, the age fifty-five is kind of magic. It is where the penalty for early retirement stops, so when you calculate your retirement, what you're going to get out of the old retirement system, which I was in, you get quite a reduction for every year under fifty-five you are.

Once you're with the government for twenty-five years, which I had easily been—remember, I was twenty-one when I went in. I hit twenty-five years when I was forty-six. Once you've been in for twenty-five years, it's your age and years of service that drive it. When you hit fifty-five, the penalty stops. Now, your retirement keeps going up each year, both based on your pay—pay keeps going up—and the percent of that that you get at retirement goes up, too. But not near as much as the penalty.

So it's like a two-slope curve. Your retirement is rising rapidly as you approach fifty-five, and then doesn't rise near as fast as you pass through. It turns out, that's an artificial age. It doesn't really matter, but it's an emotional one because the penalty stops going away. That basically goes away.

I remember, through my career, I'd always heard people say that one of the biggest problems they had when they hit their fifties is you start thinking about what does your estate look like. You don't want to be a burden on your kids, you want to make sure you've got enough money set aside so that when you do retire, or when you have to retire, which is the real fright—you know, you become disabled in some way—that you've got enough set aside.

When you look at the government salary, it kind of goes like this if you're—I always do it with hand signals, which you can't do on a recording, but if I spread my hands apart up and down by about a foot, and I say, "That's what I'm making if I stay," and if I lower my upper hand down by about two-thirds or half and say, "That's what I make when I leave," then the difference, which is moving that lower hand up above the middle one, saying, "That's really what I'm making when I stay."

The fact is that there's no incentive to stay, financial. It's all in the work you're doing and, in NASA's case, the fun of being part of pushing the frontiers, which makes everybody stay for a long time. So I think it was those three factors that were in my mind. I said, "Hey, it's time for me to change."

The only change for me would be go to Washington. No promotion, but I'm not going to be center director. There's no other job I want. That is, not computing. They would love to get Sue back in Washington, no question about it. In fact, they have her back right now, for a while, and so they'd love to move us both up there.

The locality pay in Washington means an 8 percent cut, by the way. Same grade, same pay, you get an 8 percent cut if you move to Washington, if you're a federal employee. And so there was just nothing to do except start talking about retirement, so I did. That's a matter of when and not if, anyway, right? I talked to George Abbey about it early and he said, "Yeah, I guess you paid your dues." So about a year after I talked to him, I ended up retiring, which was January of 2000, the end of January 2000.

As often happens, as soon as word got out I was retiring, I had an offer, a good, challenging offer, for another job, which I took, which is actually associated with NASA. Made me vice president of a company, one of many vice presidents, but vice president level of a company of called OAO, that does the computer services. My role involves the Code M centers, all four of them. So I'm in the unusual position of trying to not show my face very often, because people remember me for what I am and I don't speak for NASA anymore. I speak for a company. But that's been very good, it's been very good.

When I watched people leave the center and retire when I was with NASA, I always thought it was good when I saw them go work for one of the contractors in the area. I mean, as long as it wasn't swinging-door kind of stuff, that I never have liked, but if it was a notion of just switching, just like moving to another place. People move. If they're still contributing to the space program, then the space program didn't lose anything. That, I think, is healthy. That's healthy.

So that's my reasons for retiring. Nothing detrimental at all, but also absolutely no problem. I'm happy as a duck in rain. I have no desire to go back and that's always been the case. When you change, if you kind of like what you did, then it's fond memories looking back but you never want to go back there, right? No problem.

RUSNAK: Out of these many jobs and responsibilities you've had over your time with NASA, did you find any particular one was the most challenging that you faced?

GARMAN: I think the CIO job was the most challenging, and the combination of computer security and trying to effect standards. But that may not be what you're reaching for. I think it's always the most senior position you have that is going to be the most challenging, and that was my most senior position. As you move up, there's always the risk of Peter Principle, right? Moving beyond the point where you're confident to do a job. A person's reach should always exceed their grasp a bit, so I think it's normal to say that the last job I had, many of its aspects were the most challenging.

If I gave you a couple of second places, it would probably be moving the Space Station Program from Houston up to Washington. That was extraordinarily difficult. There was a lot of politics, a lot of technical issues. Whenever you create a new organization, there are many people that view it as a means for getting promoted or a means for getting out of something into something else. You get all sorts of drives and motivations that aren't necessarily connected with the success of the program. And turf battles. "I'm in charge of this." "No, you're not. I am." I don't mean me, but you get that argument going.

So that was probably my first major experience with such a wide variety of both technical and political issues, that I recall it as being a big challenge. Not unusual. I mean, nothing bad in it. Nothing unusually bad. Life is full of good and bad, but it was probably the first time I really got dived into politics outside the center, big time, because I was suddenly a Headquarters employee for a while there.

And then going back, the biggest technical challenge was the onboard operating system for Shuttle, no question about it. Apollo was built when I came on board. I learned it and became an internal consultant, so to speak. In Shuttle, I was accountable for a lot of it and it was very, very risky, very difficult to get it pulled together. Yes, that's pretty good.

RUSNAK: Through all these positions, they've obviously all been related to computing, whether it's been flight computers or ground. What is it about computers that really has held your interest for all these years? What do you find most appealing or personally challenging?

GARMAN: Well, two ends to that one. One is highly philosophical. I think that machines are what enable the human race to move forward and up, and machines have always been physical. They've been the things that make our arms stronger, our hands stronger, our feet stronger. We don't have to run; we can drive a car. We don't have to lift; we can use a tractor. We don't have to build by hand; we can have factories. Computers are the first devices that actually help the mind, and they really help us think better, faster. People say that the fact that kids are not very good at adding and subtracting by hand today is bad. I don't know if it is or not. Let's put those brain cells to work on higher-level thinking if I can have a hand calculator or a computer spreadsheet to do that mundane kind of stuff.

On the philosophical end, one interesting example was, there's a website in Australia I found that had a pictoral example of how fiberoptics works. It was done because fiberoptics is the big networking kind of thing these days. But this darn thing actually had moving pictures and it showed exactly how fiber was built, what the layers were and lights reflected through the fiber, a very, very visibly instructive and when I saw it, I sent it to a colleague of mine who used to go to MIT, and in fact, he's forwarded it—and by the way, just forwarding it, URL [unclear], right? Forwarding all the information, that's what's amazing. He forwarded it to his former professor at MIT, who has since adopted it in his class, by the way. But it absolutely struck me because the ability to visualize that used to take years. You'd have to play with it, they'd give you examples in a class, they'd give you drawings. You had to abstractly in your head make it move to really understand it. When you can visualize it, you can learn so much faster.

These kinds of machines not only enable us to do more, but they enable us to learn faster, too. It's almost a crime that they're so poor, that they're so inefficient, that they're so unusable, that they're so frustrating. So working on that, to me, whether it be on flight computers or anywhere, that's been a neat thing.

The other end of the spectrum, the much more pragmatic end, I suppose, is that they're fascinating. They're like puzzles. It's something you can conquer. You can beat them. They're always changing. The more skill you get in it, the more you have to keep the skill going. There's nothing that ages faster than detailed knowledge about some piece of computer technology. It's over in six months. So if you want to be stimulated always, then learning that technology is fun.

I've never been a good manager in it as much as I've been a good technocrat. I've always learned the technology, which has been very, very handy in a lot of ways, like knowing how to fix a keyboard that's stuck. So if you can be continually fascinated by something and challenged by it and then feel like you're also helping, whether it's helping a nation be economically viable, which computers help you do, or just helping the human race keep elevating its level of doing things and thinking, the old Maslov principle, then, yes, it feels good. It's always felt good to be in that business.

The other piece of it is that it's ubiquitous now. It's everywhere. If you're in the business of computing, you can work with people in any business. There's no limit to it. If you want to work with the medical industry, you can go in the medical industry. You don't have to be a doctor; you can be a computer nut. You want to go work in the construction industry, you can go work in construction. You can go anywhere you want because it's a very common denominator now. That's another piece of it. Of course, that's the kind of stuff I tell people when they ask me, "Should I go into computing?" I say, "Hell, yes, and here's the reasons why." But I feel it, too. That's why I've always enjoyed it.

RUSNAK: Another question. One of the points NASA has made in the past, in terms of spinoffs, they've suggested that the Apollo guidance computer is perhaps a direct predecessor of today's PC.

As someone who has worked with both and followed this whole transition, how correct do you think that analogy is?

GARMAN: I think there's a lot of truth in that, but not at the first-blush thing. That is, the Apollo guidance computer was no PC. I knew the Apollo guidance computer and it's not a PC. I think what people are referring to is the notion of getting the technology into a small space that could do the job.

Computers were, from the fifties all the way through the seventies, computers that did anything were huge. And making them smaller and making them interactive are two phenomenal changes in computing. Smaller and interactive. The Apollo computers were both.

Until the advent of the PC, computers were not interactive, not if you were a user. You handed it a card deck and came back the next day to get your output. If you made a mistake, you handed it another card deck and came back the next day to get the output. And even if you did it on terminals, it was the same thing.

The idea of having completely interactive software, a spreadsheet that if you changed the equation, all the numbers change while you're looking at it, a change of value was a change. You don't have to submit a job to get a new output. That's phenomenal change.

Well, any computer in a flight system is that way because they work in real time. They have to know what time it is, they have to interreact with the pilot, astronaut, or what have you. So putting computers into flight systems of any kind did an amazing thing to push the interactivity notion and to make them small. Apollo was the biggest thing going. It's where the most money was spent.

They're putting computers in the military. The government spends a lot on research because of the military. You may not like it, but that's the way it is. Better we spend it on research to build defense systems than to spend it in wars, but that's where a lot of Apollo technology comes from. Apollo was totally a civilian operation and a lot of money was spent. Apollo derived a lot of its computing from the military, as a matter of fact. The guidance system was out of the Polaris missile. I remember because I had to have a security clearance because of that. I mean, the inertial measurement unit was. So it's got to be philosophical. First, Apollo was real visible. Without Apollo, it still would have happened, but not as fast. Because the more money you spend, the faster things go. It's not literal.

The computers were not the same at all, but the notion of interactivity and small were certainly there. The PC was really born out of the work that Xerox did. Not IBM, not Apple. Xerox. Xerox Park in the seventies. If you read the history you'll find that even the Apple folks will agree that when they went to see the Xerox Park, they came with their ideas, with a mouse and the GUI [graphic user interface] interface and so on from there. That was totally commercial. Xerox had nothing to do with the space program.

But to take the technology that had been developed, which was the NASA, you know, from the nonstick frying pan to computing, yes, huge contribution to that. Software is the same way. Computers don't mean anything unless they have software that works.

That story I went through of the abstractional layers of computers and applications, the concept was not new to the Mac or Windows at all. People had been struggling to do it for years, but unless you do it commercially for a huge audience, it doesn't work, because you'll do it this way for one DoD project, you'll do it that way for one NASA project, and they don't talk to each other. You can't reuse it.

I think the real trick was reusable software. We reuse software all the time now. When you think about it, every time you go to the store to buy—we used to talk about reusable software and the notion of when we build something let's build it a way, in components, that if someone else needs it, they can grab the component and glue it into their program so they don't have to rebuild the same thing. And there's a lot of use for that. They're called libraries, in general. If you have matrix multiplication routines, why rebuild it every time? If you have a sorting routine, why rebuild it every time? A lot of work was done in how should you package software, so that without knowing what's in it, and you don't want to have to study the code, you just want to get it, so you create the notion of an abstraction layer where you know what the input/output is and you know what the parameters are and that's all you know and if you've ever done programming, I've just described the current level of programming, software engineering, and reuse.

That's not the way it came out. The way it came out is reuse is going down to the store and buying this piece of software, isn't it? You're reusing something that somebody else has used. Now, they built it to be sold, but what they did was take a need and build a component and sell it, either as a whole package or a series of components.

This is back to the notion that the industry had to create an architecture for that and they did. But the ideas came out of the difficulties and hurdles, largely where big money was spent, and that's in DoD and in NASA, in particular, particularly in the late sixties and early seventies. So in that context I'd agree.

RUSNAK: I wanted to give Carol and Sandra a chance to ask some questions, if they had any. All right. Well, are there any concluding remarks you want to make before we wrap it up today?

GARMAN: No. Concluding remarks are very dangerous. You end up summarizing something, and then you say, "I shouldn't have summarized that. I should have summarized this."

RUSNAK: Well, I didn't want to cut you off if you still had—

GARMAN: Thank you very much.

RUSNAK: Okay. Well, it's been a pleasure.

GARMAN: You bet.

RUSNAK: Thanks.

[End of interview]