INTERVIEWEE: BOB PATRICK

INTERVIEWER: Robina Mapstone

DATE: February 26, 1973

RM: The date is February 26, 1973, and this is Bobbi Mapstone and I am talking to Mr. Bob Patrick at his home in Northridge, California.

All right, why don't we start this where we started before, and this is going back to your education and how you got into computing.

BP: I am a graduate mechanical engineer from the University of Nevada. When I got out of school in 1951 I went directly into the Air Force, and after being sent around two or three locations, I ended up at Edwards Air Force Base out in the California Desert about sixty miles from here. Edwards at that time was the home of the flight test center where all the manned aircraft for the United States Air Force were flight tested.

I was sixty miles out in the desert and it was a womanless environment, generally. A great opportunity came up to go to UCLA on Saturday mornings and be in town for the weekend. So I took that opportunity and

entered as a Saturday morning student of Everett C. Yowell. The last time I heard of him he was still with National Cash Register in Dayton, I believe. The National Bureau of Standards had a machine called the SWAC, the Standards Western Automatic Computer, at UCLA that we looked at and pushed our nose against. We didn't learn anything about it because the course that we were taking was in the Card Programmed Calculator (CPC).

The CPC was a new, at that time, IBM machine, which married a tabulator and an electronic computing box. I suspect that somewhere in the dank history, Rex Rice had something to do with getting IBM a product out of that.

RM: Rex Rice was a user. It was actually Toben and Woodbury who did the marriage.

BP: All right. When the machine came from IBM it had all the interconnections to connect this computer box to the calculator, but there were three plugboards on it and you had to wire these plugboards before it would do anything useful. Ev Yowell and his crew at the Institute at UCLA had wired up a standard set-up for these plugboards. It was a three address CPC board, and the purpose of the Saturday course was to teach us how to program the machine, use it for useful things,

and eventually to teach us how to wire those standard set of boards so we could have a machine just like it (out in the desert). There were about forty in the class and from Edwards there was a Colonel, a Lieutenant Colonel, and myself. I was a Second Lieutenant at the time.

RM: Can we just get a date in here?

BP: This was about September, 1951. Part way through the course our CPC arrived out at Edwards. It was in three monstrous wooden crates on the same kind of low-bed trailer normally used for carrying bulldozers and heavy duty tractors and so on. My instructions were almost the classical military instructions. They were, "Lieutenant, get it off that trailer, because we need the trailer." (laughter)

We moved that computer into a wooden building and wired the Yowell boards to control it. We went down to UCLA and checked them out, and we were in business approximately the first of the year, 1952.

There was a young fellow out there who was an enlisted man at the time by the name of Carol W. Sweet and when he got out of his service tour he stayed over as a civilian to be the first professional operator on this CPC.

RM: Did you teach him?

BP: I taught him to operate the machine. This was a Model 2 CPC. There were a few Model 1 CPCs but the plugboards were entirely different than the one we had, and the features were different. One of the Mod 1's had been installed at Inyokern, California, at the Naval Ordinance Test Station about half way from here to Bishop, California. We had one of the earliest Model 2s in the United States.

Carol Sweet later left Civil Service and wandered around and ended up working for IBM as a salesman, and for the first time in twelve years or so I saw him the other day. He works for IBM and is stationed down at their Westchester building off of Marina Del Rey, here in Los Angeles.

RM: What kind of work was being done on the CPC?

BP: Well, the flight test center did the following things. We'd get brand new Air Force planes and we would install instrumentation in them to measure the fuel flow, accurate air speed, and flight conditions. The data was recorded either on oscillographs in the airplane, by cameras which photographed instrument panels, or by flight test engineers riding along and recording

the data manually. At that time I was a flight test engineer and flew in these military aircraft, running the cameras and manually recording the data.

When the senior officers, the pilots, would get back down on the ground, they would go over to the officer's club and party up. The junior officers, myself included, had to see that the data got transcribed on the standard sheets, reduced and plotted up so that it would go into the standard flight handbooks. Every pilot who was later going to fly those airplanes used data out of the flight handbooks to find out what the range of the aircraft was, what the service ceiling was, what kind of armament load it would carry, what the limiting speeds were, etc.

We had the elite of the Air Force pilots out there. Colonel Frank Everest was the head of the Air Force Pilot crew (he was an ace). Major Chuck Everest who was the first man to fly faster than the speed of sound also was one of the test pilots there. They had a bunch of us puppy dogs following along behind these guys recording the data. I worked for a Captain, at the time, by the name of Bill McGruder. Billy McGruder later became a flight test pilot himself and worked for Lockheed on the SST. He is now the

advisor for Science and Technology to President Richard M. Nixon in Washington. It was an elite crew.

After a test flight got back onto the ground, I had a crew of ladies who retabulated data and read the film that had been taken in flight. After we tabulated it we punched it onto punched cards and fed it through the computer that was run by Carol Sweet. I did the computer programming, and some of the operating. This covey of ladies did manual plotting for us and read the film and tabulated the data. These were the activities that took place and that I performed, I guess starting about the first of 1952 through the early part of 1953 when I got out of the service.

I left all this equipment there, and I went to Convair in Ft. Worth, Texas. For my Edwards tour I can't take claim for being anything but an early user of computer equipment, where the equipment was built and maintained by IBM, and the intelligence, what we would now call the software, was buried in those very complex boards that Everett Yowell and his people at the INA wired up and we duplicated.

RM: Did you know people like Dereck Lehmer and Huskey when you were going to UCLA?

BP: No, I met Harry Huskey many years later. He had been around at that time, but essentially we were just students at a Saturday course that just happened to be given next to where the SWAC was, and we were mainly concentrating on the CPC which was in a different building of the same complex.

RM: Did you become familiar with SWAC at all?

BP: We did not. All we did was to see this wild wooly thing there. It had a collator as its input and tubes and wires hanging out of it. But we didn't do anything with it.

RM: And you had no feel at that time that this was the wave of the future?

BP: No. It was a straight engineering course and we took a straight engineering approach to it. We had a job to do and this may have colored my entire life in computing. There was real work to be done, we didn't play NIM with it, or chess or three dimensional grab or any other damn thing. We had a job to do and we focused on getting that job done and putting that installation in at Edwards and using it for useful things.

RM: By the time you left the Air Force base, had they gone into a 701?

BP: No, that was way too early. You're jumping the calendar. For various reasons, family and domestic, I went to

Ft. Worth, Texas, to work for Convair. They had four Model 2 CPCs that were in the accounting department but were being programmed by engineering. This was one of these first, early jurisdictional disputes. The computers had developed out of the punched card genesis, and the accounting people still had control of them. The IBM salesman made calls on the accounting department, he didn't make calls on the engineering department. But there were engineering plugboards in those machines and we ran engineering jobs on them. They were installed for us, but the accounting people paid the operators and owned the real estate the computers were installed on.

I was the fourth man to join the Convair, Ft. Worth Computing Center, which is now one of the maybe top one hundred computer centers in the world. Henry S. Wolanski was the leader of the little group. He was the Computations Group Manager or something like that. He wasn't very bright, but he was the first on the scene so that gave him his rank. Warren E. Myer was there, and he was a CPC programmer. Warren stayed there for several years and then left and worked

for the Systems Development Corporation in Santa Monica for many years. I haven't checked a SDC phone book recently, but Warren may still be there as a computer programmer. The other man that was there was John T. Derr, who is now called Dr. Derr. He came out here, won his Ph.D. and now works for the RAND Corporation in Santa Monica as a mathematician and lives in Granada Hills, California. Wolanski, Myer, Derr and I were the first four men in that computer center.

Wolanski, Derr and Myer had never been allowed to touch the machines because they belonged in the "wrong" department, and there was a jurisdictional dispute between them. It was almost like a union dispute only there weren't any unions involved at the time. On the other hand, I had come out of the desert where I owned one of these machines, and had learned everything the hard way. Derr and Myer did the routine programming and I did all the special jobs, all the flight test data reduction, all the strange stuff. That was how the four of us first started work at Convair.

Things then started happening very rapidly. Let

me back up for a moment. The IBM salesman at Edwards Air Force Base was a fellow by the name of Don Pendery who, after he left Edwards, got quite high in the IBM Corporation running up the ladder like a monkey. He did very well. The last I heard of him he was special assistant to the head of IBM World Trade and stationed in France. I got to Ft. Worth, and I don't even recollect meeting the IBM salesman because he made calls on the accounting department, but the applied science representative was a young fellow by the name of Paul W. Knaplund. Paul Knaplund is an IBM Group Vice Presdent now stationed in Armonk.

Convair was one of the people who ordered an IBM 701 and we were destined to receive serial #7 Model 701. The serial numbers are important as to the phasing of these things. Obviously there were six ahead of us, but we weren't too far down the line.

Serial #1 701 was installed at IBM's New York Center at 590 Madison Avenue in a big inner room that I guess is now a museum or a mausoleum, or something. (laughter) Sometime about the fall of 1953, a group of us went from Texas to New York to get machine time on the 701 to run our acceptance tests, to take our programmer training, and actually prove it fact that

we knew how to run the machine and how to program for it. We went to World Headquarters, as it was known then, and ran on 701 #1.

Paul Knaplund and I were working together on a big structural problem which was intrinsic in the design of the B-58 Hustler bomber. The B-58 had a special wing spar design which was new at that time and it was one of those things where the design engineers had the right theory, but with desk calculators they couldn't compute. Paul Knaplund and I worked up the computing, automated the computing algorithms, as we would call them now, so that we could do the design for wing twist and shear on the B-58 Hustler under various flight conditions. That was the program we checked out on this #1 701 at World Headquarters shortly after it was installed.

That was in the days where anytime the machine made a mistake, the first thing the customer(maintenance) engineers would do would be to run over and close the drapes so the people walking by on the sidewalk never saw an IBM machine with the covers off. They made us wear our coats all the time, so we all looked like gentlemen. There was some EAM equipment in the corner room and while I was running this EAM equipment I jammed some cards in the sorter and I, without thinking

since I knew all about it, opened it up and went to fetch out all the mangled cards. All of a sudden I was descended upon by a whole hoard of attractive young men in dark suits and striped ties. They just let me know that wasn't to be done. They grabbed hold of this machine and they hauled it around a corner behind a blind so that nobody could see us take the mangled cards out of this sorter. (laughter)

After we checked out those programs and returned to Texas, our machine came. IBM had had some trouble installing 701s. A young fellow, I didn't know him but I thought he worked for the mover, came out with this machine. He was a heavy set fellow and he had on a tee shirt and a pair of old baggy pants and was helping install that machine and lugging things around. Hell, we thought he was one of the moving company. We now know that that man was not one of the moving company men, he was one of the IBM engineers. His name was Bob Evans. He is now President of Systems Development Division of IBM Corporation, stationed in Harrison, New York. Bob Evans helped install 701 #7.

We didn't know much about physical facilities then and we had an open circuit air conditioning system. The air conditioning chilled the air and fed chilled

PATRICK

air into the bottom of the machines. The heat coming out of the machines heated the air and then the air was exhausted into the room. If all this worked well the room was comfortable and the machines were cool. During the early days of that machine (before it was even turned over to Convair) the IBM engineers didn't trust leaving the power on while they went out to lunch, so they turned the power off the machine without killing the air conditioning. When they came back, there was frost on the insides of the room, the whole machinery was a temperature of about 40°, just about like a meat box. We had all kinds of wild troubles. (laughter)

Finally the machine got settled down and turned over to us, and things were humming along reasonably well. Convair did some reworking of the electrical power somewhere feeding the building, and when they did they reversed the phasing on two cirucits on the massive three phase power that fed the machine. We came in the following morning and punched the Power On button and blew up the goddamn machine. We were down almost two weeks. Literally a bushel basket of ruined diodes came out of that machine. Since then, and I think we were one of the main reasons, the IBM power sequencers all have protection circuits on them.

PATRICK

You can't do that to an IBM machine anymore, they just refuse to power up if the circuits are screwed up. We learned everything the hard way.

Finally the machine got settled down and it performed useful work. As I recollect we didn't trust it more than thirty minutes. If we had a job running more than thirty minutes, we had to put in a restart or two. We tried to put in a restart every fifteen minutes, so that if the machine did break down we didn't lose more than fifteen minutes worth of computing.

The machine ran well for awhile and then it started having intermittent troubles. They found that it had what they called cold solder joints. The machine had been assembled with manually soldered circuits in it. The pluggable circuits with the tubes in, could be removed and exchanged around. The back panel wiring had some circuits that had not been heated up enough for the solder to flow around the pin and the copper wire.

Our set of maintenance engineers, as I recollect we had three full-time maintenance engineers, couldn't seem to get the machine fixed. They sent down an older fellow, he must have been all of 40 because we were all very young, named Paul Bumgardi. Paul Bumgardi took that machine for an hour a day, and he and his two CEs resoldered that entire machine in the field.

Every morning they'd do a few solder joints, a hundred, two hundred or something, and in about a month they had resoldered the whole machine and the problems went away.

Paul Bumgardi is now stationed in Poughkeepsie, New York where he has been for several years. He is a senior technician in IBM's new memory development, or was the last time I saw him, where brand new memory designs are put together. Paul leads the crew that builds a brand new memory design before it gets into a product.

The 701 machine now continued to move along pretty well, and as I recollect somewhere in the fifteen or so months I was acquainted with it, we took an upgrade and put on another 4,000 words of core. It had 4,000 words to begin with and I think when I left we had 8,000 thirty-six bit words on it.

RM: Of core?

BP: I'm sorry. Pardon my mouth. Those are CRT Williams Tube memories, electrostatic memories. I think we put on a second box.

RM: Did IBM do this?

BP: Yes, we ordered another box from IBM and went from 4,000 words to 8,000 words if my memory serves me correctly.

RM: Did you write your own programs? Or did you use IBM programs?

BP: There were two sets of software involved then, and we didn't even know it as software. IBM put out an assembly program, which as I recollect was very slow, and a set of utilities. Johnny Derr, Warren Myer and the people working with them were doing very, very large simulations and they were programming in assembly language. IBM also put out an interpretive system which was called SPEEDCODE. SPEEDCODE was the product of a whole crew of guys, but two of the guys on the crew were John Backus, of John Backus Normal Form and Formal Languages fame, and Ted Glazer. Ted Glazer was blind then and he and his dog worked on, among other things, the tracing part of SPEEDCODE. This allowed you to trace and print out the process of calculation. Ted is head of the Computer Sciences Department at Case Western Reserve University now.

RM: He's someone I really must talk to.

BP: Well, he is a very interesting guy and just sharp as a whip. They put together the SPEEDCODE system which was a three address, floating point system which was

very similar to the three address, floating point CPC boards Ev Yowell had invented and which we were using a derivant of at Convair.

Since I knew the CPC, I was assigned to do the transition of programs from the CPCs into SPEEDCODE. Using EAM techniques and using the CPC itself, we did a kind of a transliteration without redoing those programs or re-checking them out. We did most of the transliteration with the punched card accounting machines changing the formats, and then I went in and hand tailored them so that they ran. I was the local SPEEDCODE expert because my background with the CPCs made it a natural for me to make that transition. So there was the first generation zero to generation one.

Things grew up at Convair and were going along pretty well, and lo and behold we had to formalize things a little bit. When we first started, each programmer ran the machine himself, programmed it himself, and if he required only a few cards, he keypunched them himself. If he required a lot of cards, he sent them over to be keypunched in the accounting department. Later we found that we had to have some write-ups (because you couldn't remember what all the stop codes stood for and what all the switch settings were). We had to introduce some write-ups and I wrote some of

what we would now call programming standards for how to put the required information into a write-up.

The time came when we couldn't get all the work done in one shift plus a little over-time and we had to open up a second shift. Shortly before that time we had to put on a professional computer operator. Once we programmers got things to where they would run, the professional computer operator would feed the data to them and watch after the care and feeding of the 701 to get the answers out. The programmers could then go back to programming. The first professional computer operator at Convair just brought you a cup of coffee. Corinne was the first professional operator at Convair (later my wife) and she ran the production jobs at Convair while we went back to programming the other stuff.

When the time came to open up the second shift, we hired a fellow as the second shift production operator. I transferred from first shift to second shift, temporarily, to train him and to get the second shift production running. I worked second shift for a month. Corinne and I weren't married at the time, so I saw her after I got off work, you know one to two o'clock in the morning. She didn't get much sleep during that month. (laughter)

PATRICK

After the second shift was set up and humming along, and I had worked for Convair about fifteen or sixteen months or something, I got tired of doing military work and figured that I would leave Texas. I wanted to continue in the computer field, but I didn't want to do military work. There was a guy who had offered me a job to be in charge of a 650 installation and I couldn't make up my mind whether I wanted to be a big fish in a little pool or a little fish in a big pool.

Finally I made the decision and moved to Detroit and went to work for General Motors Research. At that time they had a Model 2 CPC, just like the one I had been working with for three years or so, and they had on order serial #17 701. I went to Detroit in 1954, maybe mid-1954 if my memory doesn't escape me, and helped them move into their new Tech Center, which at that time was a brand new facility out north of Detroit, all stainless steel and glass. Shortly after we moved into the new Tech Center with the Model 2 CPC, our 701 came. I helped bring them up on the 701, and helped introduce SPEEDCODE to the 701 there. I started doing gas turbine applications work, which was a natural follow on to the work I had been doing at Convair. (I had been doing some gas turbine work at Convair.) Then I did some detailed

gas turbine predesign calculations, logical simulations and sizing calculations for General Motors who was then building experimental motive gas turbines. They had a car called the Fire Bird, which was a race car with a gas turbine in it. They put turbines in trucks and busses and so on. So in 1954 and 1955, I was supporting the gas turbine design work at GM. When we got engines built, I then started doing gas turbine data reduction using the CPC computer. When the 701 was installed we transferred our work, again I helped in the transition from the CPC to the 701.

That brings us up to early 1955 when Corinne and I got married. My long distance phone bill was so great between Detroit and Texas, it was cheaper to get married. That ends my early part in the computer field. Up to that point I didn't do anything very spectacular. I was a user. I was a journeyman programmer. I broke ground as far as production operations, documentation, second shift operations and so on for some very early commercial machines. But again, as I indicated earlier, the emphasis all the time was a straight engineering approach. We had work to be done, we weren't fooling with the machines, we weren't playing around with them. I didn't know anybody in those days that played Nim, or chess or checkers or did any of this foolishness.

We just had so much work to do that we concentrated heavily on pushing it through the machine.

RM: When did you get onto the West Coast?

BP: Well, that's the next phase. I think we've finished phase one. That's phase two. (pause)

I entered the second phase of my adolescence in computing while employed by General Motors in Detroit. When the 704 was announced, General Motors had one large computer in the entire corporation, and it was the one that I worked on in Detroit. The head of the group was a fellow by the name of Donald E. Hart, who is still there, and there were four group leaders answering to him. Ed Jacks, who is still there, George Ryckman, who is still there; Pat Hayes, who moved to General Motors over here in Santa Barbara, California; and I was the fourth one.

When the time came to plan the transition from the 701 to the 704, each of us was given some responsibilities. As I recollect Ed Jacks was given the responsibility for the mathematical subroutines that we needed, because there were no libraries which came with the machine. I've forgotten what Ryckman and Hayes were given responsibilities for, but I was given the responsibility to determine the operational mode of the 704.

Now, I'm going to take us back in history for a

PATRICK

little bit. There were several special features of the SPEEDCODE system that would allow you to keep programs on drum memory, hence you didn't have to load your program every time. It's what we would now call a resident data set. There were also some features of the SPEEDCODE system that would allow you, while you were reading cards, to store the program away on drum memory so that if you needed it again you could use it without reloading it from cards. Rather than to load it only into memory, you could put it both into memory and on the drum. I had been using these features in a very special way: I found, I could get on and off the machine before anybody else was ready to get going. General Motors was block scheduling their machine (as Convair had) and I found that some of my mathematical friends would walk into the machine and they would address it and say, "Where am I, what should I do first?" If I was really ready to go, I could get a thirty second shot while they were getting set up. So I was sneaking on the machine, getting a quick test shot, and getting off. Due to this technique, my work was moving along very rapidly.

For my conversion tasks, I was given the assignment to study, think about, and come up with ways to run the 704, and to operate it. I later found out that there

PATRICK

were some people working at MIT, I guess it was part of the Lincoln Labs at that time, and they were thinking along the same lines. But I swear to God, I didn't even know that Lincoln Labs existed. I was an engineer working for General Motors, I had an assignment to do, I had some success in making quickie test shots myself, and I thought if I can take my successes and export them to the remainder of the professional staff we can all get a lot more machine work done in one day. Machine time was a very critical resource.

I did what we can call a classical industrial engineering time and motion study, except that I was a computer technician. Out of this I came up with what we called them (and for maybe five years) the three-phase operating system.

In the first phase of the operating system, you loaded everybody's jobs into the card reader, read cards like crazy, translated the cards into binary, did whatever edits were necessary, and wrote them on a magnetic tape. During the second phase you read that magnetic tape and did the computations, the compiles, the executions or the assemblies; and the tests wrote their outputs on a magnetic tape. With our 704 came two special purpose computers. One read cards and made magnetic tape, and one read magnetic tape and printed

lines. The numbers I recollect are the 714, which was the card reader, and the 716, which was the printer. If you used one of these to make magnetic tapes, then the main computer was faced with a steady stream of input tapes being hung and a steady stream of output tapes being carried back out to be printed. This was the first instance, to my knowledge, of smooth work flow in and out of the computer shop being designed by an industrial engineer. It was possible to overlap the setups so that you didn't have the machine ever waiting for set-up. One magnetic tape was ready while you were reading from a previous one. As soon as you finished reading from the first magnetic tape you manually switched to the second magnetic tape and the main machine kept right on computing. I had designed all of this on paper. If you are interested I may be able to find the original design documents in my archives in the garage. I made a write-up on this and I presented it to the SHARE #3 meeting in Boston in 1955. I described this operational mode and solicited comments from the SHARE membership. I do not remember the date.

RM: SHARE was formed in 1954, I believe.

BP: I believe that is correct. I was at SHARE #1 in Philadelphia or maybe that was SHARE #2.

RM: Wasn't SHARE #1 on the West Coast?

BP: The West Coast guys had been working on an interpreter called PACT. Out of that cooperative effort they thought that a nationwide organization would be useful, and the names that come to mind are Blair Smith, Paul Armer, Jack Strong, and Frank Wagner. I don't remember who we'd really call the father of SHARE, and I don't know where the SHARE meetings started to be numbered. So I'm not real sure where SHARE #1 was.

RM: Someone has called the IBM guy, Blair Smith, the father of SHARE because he picked up the tab for that first meeting.

BP: Yes, that's right. Blair Smith was the IBM applied science representative assigned to RAND as I recollect. He picked up the tab for the meeting and some of these things got organized. I was at a SHARE meeting in Philadelphia or Pittsburgh and I thought that was SHARE #1. But my memory may be faulty. SHARE #3 was in Boston, and I'm pretty sure that was the meeting where I introduced this operating system.

At SHARE #3 was a fellow by the name of Owen Mock. Owen is now with Computer Sciences; prior to that he was with North American. Owen had in mind an operating motif for the North American Computers; he was thinking about it, he had some notes written down on it, but he

wasn't ready to present it yet. When he saw what I had in mind, he thought that my ideas were better than his so he said lets jointly go together and produce it. When we finally got down to the discussions after SHARE #3, we found we could agree on <u>almost</u> everything.

There were really two versions of the operating system we produced. The front end was the same, the back end was the same, and they differed only in the middle. The North American Monitor system was the name of it out here, and it was called the GM I/O system in the East. I guess there may have been fifteen or twenty locations that used it. We were in competition with a system that Don Shell put together at GE Evendale. His system never got exported very far and ours became used quite a few places. Ours then became the basis for the SOS system, and the SOS system, in turn, was the basis of the IBSYS system, and the IBSYS system was in turn the basis for the IBM 360 system called TSOS--tape operating system.

RM: What was IBSYS?

BP: IBSYS was the IBM operating system for the 7090, which the entire scientific community used almost without exception. If you were drawing a tree, it would come from the SPEEDCODE experiments that I had practiced, to

the design that we did for the General Motors I/O system. It would fork and there would be the North American version and the General Motors version, which were two forks off the same stem, and then we would go from these two and recombine into SOS, which was the SHARE Operating System, and then IBSYS. From IBSYS we went to the Direct Couple and from Direct Couple we went to OS360 and ASP.

RM: Okay, so really what happened was it started with IBM, it spun out and it came back to IBM. SPEEDCODE was the beginning.

BP: That's correct, SPEEDCODE was the root of it. Most SPEEDCODERS didn't think about using it the way I used it, but the features were there and they wouldn't have been surprised. However, IBM didn't sell it that way.

Now back to the joint NAA/GM effort. Owen Mock felt that my ideas were worthwhile, we talked jointly and found that we could agree on most of the system, and we immediately scheduled a series of meetings. I brought a crew out to California and we all sat down and decided on the specific definitions of the interfaces between the three phases, and who would do what. The first phase North American did in its

PATRICK

entirity and gave to General Motors, including all documentation; the third phase General Motors did in its entirety and gave to North American; and we each did our own special versions of the middle phase, the execution phase, the area where we couldn't agree. The three phases were input phase, execution phase, and output, and we had two versions of the execution phase. The first phase and the last phase of the systems were absolutely identical. We exchanged binary decks and updated them together.

A history would show that Owen and I should have worked together longer on that execution phase. I had some ideas that were clearly superior to his, he had some ideas that were far superior to mine, and if we'd sat down and bashed our head together we probably would have combined those ideas and come up with just one thing. When we went to go to SOS, they picked some of my ideas and some of his ideas plus many ideas that neither he nor I had. We gave up too early on that center phase.

We built the system and we installed it at North American and at General Motors Research in Detroit. We each made our system available to other people and as I indicated earlier, it was used in about twenty places around the U.S.

RM: Was it distributed through SHARE, or did people come directly to you?

BP: I think the comments and letters and so on were passed around through SHARE, but at that time SHARE wasn't nearly as organized as it is now, and SHARE correspondence was mainly a way to find out who had something so you could write directly for it.

RM: I see. As opposed to an organization that was generating its own systems.

BP: That is correct. IBM now helps pass some SHARE programs out with their Type 3 program distribution. You can contribute programs to IBM, call them Type 3s, advertise them through SHARE and get them distributed. IBM pays the distribution costs. But in those early days if you wanted stuff, the best way to get it was to get in an airplane, or in a General Motors car, come see us, and carry your copy away in the car trunk. (laughter)

RM: What kind of projects was General Motors doing with their 704?

BP: In addition to the gas turbine work which I was doing, Ed Jacks and his people were doing a lot of suspension work: springs and the dynamics of automobiles. We weren't doing very much engine design work, but we were doing a fair amount of engine data reduction work. We

PATRICK

had test cells all around the test center, and we did the data reduction of the data collected from them. We did some missile work, not very much, because we were generally associated with the car divisions. Allison Division of General Motors came up to run our machine until they got their own, and they did aircraft gas turbine, aircraft turboprop, and propeller design work.

Somewhere in through there, and I've forgotten the date again, the United States government decided to prepare a ballistic missile. They had a ballistic missile program at that time called WS105A, which many people out here know as the Atlas Missile program. It was a long range missile program and the guidance was being done by Lincoln Labs at MIT. The government wanted an intermediate range missile, they wanted it quickly, and they decided they were going to do it in one year. North American Rocketdyne Division had the liquid fuel engines that were ready, the systems contractor was Ramo-Wooldridge, which was at that time down in Westchester (LA) by the airport, and the guidance contractor (that's the other big chunk) was the AC Spark Plug Division of GM in Milwaukee, Wisconsin. AC Spark Plug had no computers and my team did all the computing for that project from Detroit.

RM: I was going to say, that's quite a thing to do a guidance system without a computer.

BP: Yes, AC Spark Plug had been looking over the shoulders of Lincoln Labs for many years. They all had badges and they ran back and forth to Boston, because they were the manufacturing facility for the Lincoln Labs designs. This was the first design they had to put out for themselves. Draper of MIT provided the concepts and so on for the design, but AC Spark Plug had to do it.

I got a call one afternoon and they sent a Cadillac limousine by my house for the suitcase my wife packed and they picked me up and put me on a corporate airplane and they flew me to Milwaukee for a meeting the next morning. That was the meeting which split the WS315A program, which we now call THOR, from the WS105 air missile program, which we call Atlas. At that meeting the project was parceled up and I was speaking for the General Motors Corporation as to what computer capacity we had. I went back that night to Detroit and I got my pick of the computer people in General Motors. I could have anybody I wanted to. Dick Sullivan, Floyd Livermore, Betty Jean Haroff and Randy Hansen were the five.

PATRICK

We all had beautiful offices with windows overlooking the lakes and the fountains and stuff, and they moved us out of those beautiful offices and put us down in a basement office which had no windows at all and put a guard at the door. We were the only classified island in that whole facility with an armed guard at our own door.

We worked that way for eighteen months. I worked four days a week in Detroit and one day a week in Milwaukee. We put out the computer programs to design the guidance system for the Thor. We had some very good mathematics done by Dick Sullivan that in some ways hasn't been equalled yet. It's better than some of the stuff that the RAND Corporation did ten years later, as a matter of fact. It did automatic optimization in several dimensions using the computer. We then did simulation programs and they launched the missiles in test mode from Cape Canaveral using our computer programs. They later shipped the missiles to Europe using those same programs. The missiles went on station in Europe using the guidance computer programs we developed in the basement in Detroit. That was a big chunk of what that 704 did. The machine came just at the right time and we really put that mother to work.

That was five of us out of a crew of maybe forty which was the size of the whole staff at that time. The rest of the guys continued to work on automobile stuff, but that little nucleus of us were working on very high priority military stuff. We had everything we wanted, except we didn't have any windows. That was the only thing we wanted that we couldn't have. I guess maybe thirty percent of the work that was done on that 704 machine at that time was getting that guidance system designed, the missiles ready to go, and then getting ready to ship and go to station.

The other people were doing some experiments in operations research and mathematical modeling which was the big interest area of this fellow Pat Hayes I mentioned. When we finished up with our missile work, they had a big mathematical model which they wanted to build, but they didn't have any programmers. This project had reached the attention of a couple of Vice Presidents of General Motors. It was a system to schedule the soft trim inventories for the Buick motor car. Soft trim consists of headliners, door insides, rugs, seat covers, all the stuff which is made out of cloth and must be dyed to match the color of the car you are putting it in. It is assembled in trim sets and obviously they are different for two

and four door cars, and convertibles and so on. Hayes had found a way to improve the scheduling of all these soft trim materials for Buick motor cars. In the pipe line at that time, for just the Buick Motor Car Division, was two hundred and fifty million dollars worth of inventory. So it was worth doing some work to improve the scheduling.

Hayes and his guys had done all the mathematical development work, they had all the concepts down, they knew what they wanted to do, but there wasn't a computer programmer among them. So when we finished the missile job, we went from the sublime to the ridiculous. We took the same five guys straight off this missile job and plunged into this simulation of the soft trim inventories for the Buick Motor Car Division.

RM: Did you at least get an office with a window?

BP: Yes, we got an office with a window at that time.

RM: I should just try and pin down a date on the Thor work, approximately.

BP: It was finished up at the end of 1956 or early 1957, I believe.

We did this giant simulation and with the simulation we could feed in car orders for Buick motor cars. We had several algorithms for how to schedule soft trim inventories (it was a multiple level inventory).

PATRICK

With several different competing algorithms, we could see, when faced with this stream of car orders, which inventory algorithm served the customers best by determining which one delivered most cars when the customers wanted them, and which one put the fewest orders into back-order status waiting on soft trim.

They had worked out all the mathematics and we programmed that up using the established crew. I'm kind of ashamed that I didn't write anything about the way we worked then because it would have been interesting to pull that out and compare it with some work which is being done in what we call Chief Programmer Teams. Recently (1973) there has been quite a bit of attention in IBM and without, in Chief Programmer Teams, where there is one king pin programmer who is a notch more senior than the rest of the people, and a small group of people, each of whom has a specific assignment. Some are assigned to document, some assigned to desk check, some assigned to write subroutines and so on. If we had documented our methods, I think it'd show a lot of similarity to what we're calling today, Chief Programmer Teams.

Although we were working in assembly language, we each had a work station. You would work at a work station for a week or so and then you'd move to the

next work station. We rotated assignments throughout the work stations. We were just delighted, because by programming in this manner one guy did the initial layout, one guy filled in the detail, one guy did the desk checking, and one guy did the documentation, we reached the point where we were able to keypunch, assemble and test five hundred assembly language statements at one time, and have them right the first time. I think that was kind of a milestone at that time. But we were mainly interested in doing the jobs and we weren't interested in the management things that the field is more interested in now, so we didn't write up any of this.

RM: That's too bad.

BP: There was one guy on this team by the name of Floyd Livermore who was kind of an introvert, but he had one magnificant ability: he could concentrate like no other human I've ever seen. When he would sit down to do desk checking it was just like pulling a bell jar down over him and sealing out all the noise and smoke. There were just no distractions when he started concentrating. He would start thinking like a computer. He'd get buried in a program and if you had multiple switching paths, he'd trace out those paths. Finally he'd surface and he'd say, "You are right," or he'd

surface and he'd say, "I think that we ought to look at so and so and so and so." When he'd say that you knew damn well you better look at it because he'd found an error. He was just magnificent. I've never seen anybody who could desk check like he did.

We had five people who were fairly well matched as far as terperament went. We were all equally bright and, although I was a little more experienced than they were by two or three years, which was enough to give me a lead, it was not enough to where I dominated them. We worked just beautifully as a team. It was an office team, we didn't socialize with one another, except for a little beer once in a while, but there wasn't a big family get together. When we came to work we had a job to do and each one of us had an assignment. It was, I guess, like the team spirit that a basketball team would feel--esprit de crops. Everybody hated us because we had priority on the computer. Secondly, when you've got a high esprit de corps and you kind of walk with a bounce in your step, everybody would just as soon tear you down. They were never successful in doing that though.

PATRICK

After those two big applications the team disbanded and I didn't have anything to do with the software. Two of the fellows who were on the original development team for the GM-NAA System were Jim Fishman and Don Haroff, were what we today would call systems programmers. They had been doing applications before, and although they weren't very good at applications they were whizzes at doing software. They were part of the North American-General Motors Development crew. After we got the I/o system up and operating they stayed with it. I met one of them at a SHARE meeting a few years ago, and he is still in software. They got into software in 1954, and they are still marching along and doing those same things. I believe Fishman is still with the company and I'm pretty sure Haroff is.

General Motors had an interesting policy then, they tried to hire only people who had deep roots in Michigan. That way they controlled their turnover in their professional organizations. Most everybody had aunts and uncles and brothers nearby. I was the only outsider.

RM: Twenty generations of family.

BP: That's right. You might go to work for another car company, but you weren't going to leave Michigan. These guys, I suspect, are still with the company.

I guess that brings us up to 1957. Okay, for the record, I worked for the General Motors Corporation from June, 1954, to August, 1958. I got out of the Air Force in February of 1953. So between February, 1953, and June, 1954, I worked for Convair in Ft. Worth. I left General Motors in August, 1958, and that's when I went to work for CEIR in Washington, DC. I worked there for about eight months before I left to be one of the founders of Computer Sciences Corporation.

RM: I think that is the story we should get into.

BP: Okay. Those are the exact dates.

RM: That's your military clearance?

BP: That's the application for military security clearance. They're known as PSQs, Personnel Security Questionnaire, and most everybody you'll come in contact with, I think, either has or has had a clearance. That's information that they all have.

RM: Yes, that's a good document to know about.

BP: After the I/O System work, I didn't have anything to do with software at General Motors. I then went off to work on some magnetic tape controlled tooling. General Motors was interested in exploiting some of the research that the Air Force had funded and had been carried out at MIT.

RM: Was this like process control?

BP: No, it's not quite like process control. You know what a drill press is? In the automobile business there are plates on which you are going to mount things. Let's say the back of a clock. It will have five little holes, and two big holes spaced for the spring mechanism, and it will have six evenly spaced holes where the clock face is held away from the back. The way that clock plates gets made is that it gets placed under a drill press, and you have to mark and drill those holes.

Some years earlier, maybe in 1954 or so, the United States Air Force gave some money to MIT to start working on coupling a computer to a machine tool. They had coupled it to what they called a milling machine. A milling machine has a shaft that comes down and you put a cutter in it, and a table moves back and forth carrying the workpiece. You clamp a workpiece on the table and it will allow you to cut grooves and the grooves are the size of the tool you put in the quill. The same technology would allow you to make a cutter that ran horizontally and to shave big blocks of metal so that you could contour it into wings, and that was, in fact, exactly

what the Air Force wanted to do with this: Contour mill integrally-stiffed aircraft wings from solid ingots.

At that time the Air Force aerospace industry was in the process of designing what they called wet wing airplanes. All our airplanes prior to that time were dry wing airplanes. They had a rubber bladder in them, and they put the fuel inside the rubber bladder. When they built the wet wing airplanes, the wings were too thin to hold a bladder and any fuel. So they had to build planes where the wing itself was the fuel tank. They had to get down in these wings and they had to leave the structure there and to cut out in between the ribs and the spars. On the outside they had to contour this block of metal so that it looked like an air foil. Of course, you couldn't do this by hand satisfactorily, so they married a special purpose computer to the machine tools to do this work. That had been successfully done, and I guess the early models of the North American F-100 were some of the first airplanes to be built that way.

General Motors was interested to see if they couldn't adopt that technique to the automotive

business, and I was the leader of a project to understand tape control led tools and to see how to bring these techniques into the automotive industry. I worked on that for awhile and I worked on what they called die sinking.

When you make an automobile fender, there is a massive block of metal which has been carved out to the stage of a fender. You place a piece of flat plate of metal in a press and a big mating die comes down with tremendous hydraulic pressure and lo and behold, after a couple of these pressings, a fully-formed automotive fender comes out. Die sinking process was very expensive and cost a lot of time. It took thirty or forty weeks to build sets of these dies, so essentially you had to release the next years model for manufacture to get the dies built, before you could see what the competition did this year. That was a very important competitive edge. If we could just reduce the time, to hell with the cost, it was the time that was very important so that we could see what Ford announced, say in 1959, before we had to release the tooling for the 1959 General Motors cars.

I worked on those projects for a while and then the projects got transferred to a different development area. They had a big jurisdictional fight between

two Vice Presidents and I was the pawn. I was in research, and the process development people had a Vice President who thought this was a process development area and I was intruding in his area. They had a big fight and they finally gave my project to process development guys, but they didn't have any computer people. They fooled around with it for three or four years, and it finally came back to research where the computer knowledge was.

In the interim, I got fed up with them giving my prized project away and after doing a couple of other jobs for them, I left General Motors and went to work for CEIR.

At that time (August, 1958) CEIR was the only computing service bureau that had a big machine. There were a lot of small service bureaus around the country, but CEIR was the only one that had a giant computer. They had a 704 installed and shortly after I got there we installed a 709. I was Deputy Director of CEIR for the year or so I was there, before I went to work for Computer Sciences.

I mentioned earlier that I was on the SOS committee. I never attended a meeting. I was invited to be a member of the committee due to the I/O system that General Motors and North American put together. My boss gave me permission to join the committee, but

about the time the committee was getting organized the government decided to produce the THOR missile and I got assigned to lead the crew. Before I could even attend a first SOS meeting, I had to decline. I carried on several conversations with Owen Mock, who was also a member of the committee, and I said, "Owen, I've got to go do this military job, here's all of my ideas, go." Owen went on to be an active member of the SOS committee.

So I didn't do any more software. I did a steady stream of applications; some military, some civilian, some computer management operations, some standards, until I got to Computer Sciences in 1959 and we did a business compiler for Honeywell called the FACT Compiler.

That was the project that started Computer Sciences.

RM: What was that compiler again?

BP: FACT, which stood for Fully Automatic Compiling Technique. That's an acronym whipped off by a guy who should remain nameless. (laughter) The first three employees of Computer Sciences were Fletcher Jones, Roy Nutt and I.

RM: Was this an offshoot of another company?

BP: No, we started it from scratch. Fletcher, Roy and I were the first three employees. Despite everything you might have read any place else, I financed the company. Fletcher didn't have any money, Roy didn't

have any money, and I had received several thousand dollars worth of General Motors Bonus in the form of General Motors stock. I took that stock to a bank in Arlington, Virginia and got money for it and that's what formed Computer Sciences. The myth of Fletcher starting it on a shoestring with a hundred dollars is just so much bullshit because we started with two thousand dollars of my money. Later, when the corporation got going, I got the two thousand back. As things fade into history, people's memories get a little shorter than they should be. Fletcher, Roy and I started CSC. Roy had all of his money tied up in a house in Glastonbury, Connecticut, and Fletcher didn't have any money at all. He was a wheeler dealer. The General Motors stock put Computer Sciences in business.

RM: Where was your company formed?

BP: Fletcher lived in Columbus, Ohio, I lived in Arlington, Virginia, and Roy Nutt lived in Glastonbury, Connecticut. We had met at many SHARE meetings and we had talked about how we would like to go into business for ourselves. We didn't know each other except by reputation and we had never done a job together.

PATRICK

Fletcher found the contract from Honeywell, and he came down and told us what we later found was to be an untruth. He said he had this contract in his hip pocket and he wanted two guys to come help him carry it out. He wanted Roy and me to go along with him. Under those conditions, Fletcher took 70 percent of the stock, and Roy and I each settled for 15 percent apiece. Later, we found out that Fletcher didn't have that contract in his pocket. A fellow by the name of Richard Clippinger at Honeywell had fallen in love with Roy Nutt and his SAP assembler for the 704. Roy had done a symbolic assembler that was very good and is the granddaddy of everybody's symbolic assemblers. Roy had done that while he worked for United Aircraft in Hartford, Connecticut. It was a very good assembler. It was fast and it had a lot of nice features. It was Roy Nutt that Dick Clippinger was trying to get his hands on. Clippinger didn't know me from Adam's off ox, and as far as Clippinger was concerned, Fletcher was a salesman. He wanted Roy Nutt because Roy had done a good job. I probably got the amount of stock I deserved in the original split, but Roy got short changed because he was the guy that made the contract a reality. We got the company going and we worked a while in Boston to write the specs on this compiler.

RM: Was this based on the SAP?

BP: No, we designed it from scratch. We were just three bright guys who were willing to work real hard. We were all barely thirty. We did the conceptual design working at Honeywell at Wellsey, Massachusetts for about three months, as I recollect. Then we sent moving vans by to pick up our families and moved the whole damn thing to California which is where we wanted to be. Fletcher was going to be the salesman and the office manager, I was going to be the project leader, and Roy was going to be prima donna designer because that is what we thought everybody was best skilled to do.

Well, Fletcher had other interests at that time, non-business is the kindest way I can think of these things, and he wasn't doing his job. Roy never had disciplined himself to have good work habits. He'd come to work in the middle of the morning and you couldn't get a quorum together until it was time for lunch. The rest of us who had children had to work rather regular hours. Roy would come and go. Sometimes Roy would come to work and he would go out to lunch and he didn't have his wallet with him. While Roy was the prima donna designer, and very good when he designed, he had a few problems with his work habits. We hired

some other people who were good journeymen and the way things were going at that time, Roy contributed about a third of the design maybe, I did about a third of the design, and the other guys did about a third of the design. So Roy didn't really shine as the star everybody expected him to be.

When we found that Fletcher was doing things with the finances he wasn't supposed to be doing, and there wasn't enough money to make the payroll as there should have been, Roy and I resigned and Honeywell cancelled the contract. It was the only contract we had at the time so all of this was very serious. Roy and I had drawn up a partnership agreement. We couldn't tolerate Fletcher and his morals, business and otherwise, so we were going to dump Fletcher. According to our lawyer, as soon as we got loose from Fletcher we could then go into business with this partnership. Honeywell told us that any two of us could have the contract, and Roy and I were going to get loose of Fletcher and have it.

We were all in Boston for a meeting with Honeywell. They caught one airplane and I caught another because I was still being project manager and I had some parcels

to deliver and explain and so on. So I came back on a later airplane. I found out that they never took their first airplane. Fletcher made a deal with Roy, and Roy and Fletcher went back again and suddenly I was out in the cold. Roy and Fletcher owned the company. So there ain't nothing I could say. I don't know what I could have done different. Roy got my stock out of the deal. He ended up with 30 percent of the stock and Fletcher held 70 percent. Of course, they had to buy me out. The cash value of the company wasn't worth much but the opportunity was worth quite a bit. They paid back all my loans and they bought me out at the book value of my stock which, as I recollect, was about a thousand dollars. So there I was on the street with my three thousand dollars; two thousand I had loaned them and the thousand dollars from my equity. I had drawn all the wages that I had coming and they had the company.

RM: So they pulled it up by its bootstraps.

BP: They continued working. They were late and in trouble on the contract. At one time Honeywell had maybe over a dozen computers in the field that they had built and installed, but since they promised the customers they didn't have to pay any more on them until they got the FACT compiler, they weren't on rent.

RM: Oh, I see. Computer Sciences was providing the software for the Honeywell machines.

BP: Honeywell was building the H-800 computer and Computer Sciences was building the compiler on contract. Honeywell had several systems installed that weren't yielding any rent until the compiler was delivered, and it was a long time late, maybe a year.

After I left Computer Sciences my first client was Honeywell. They contracted with me as an individual consultant to prepare the acceptance test cases for the FACT compiler. (laughter) So for Computer Sciences to get their compiler delivered they had to satisfy acceptance test cases I wrote and shipped back to Honeywell.

RM: I'll bet you wrote some pretty stiff acceptance tests.

BP: At that time Honeywell was competing for some banking work, so I wrote a complete application for a bank called Demand Deposit Accounting. That's the name of the application that handles your checks. I wrote a complete Demand Deposit Accounting application in the FACT language. A series of maybe ten or twelve separate computer programs that together comprised this system. I designed all the forms and I did all the documentation of it, as a matter of fact, I've

still got a set of it somewhere. That was the acceptance test I produced for Honeywell. It was a complete application which would be a demonstration package to the banks they were trying to sell after the compiler worked. They used it as an acceptance test for the compiler because it had input routines that edited data and wrote exception reports in the printer, the good data was written on magnetic tape. There were magnetic tape read routines that created files and updated files using the data tapes created in the previous process. It was a whole computer activity, now we would call it an applications system, to get them into the banking business. They used those programs as the acceptance tests for the FACT compiler.

I guess that brings me up to the other end of your history period. Computer Sciences was formed in 1959 and I finished up that work for Honeywell in probably 1960 or 1961. That takes me through the time period of interest from late 1951 through the 1958 time frame that you were interested in.

RM: Looking at our outline, I see under program language architecture that we haven't discussed Flomatic. Not only that, but I'm not familiar with it, as opposed to FORTRAN and PL/1, which I know.

BP: There was a lady who during all these times worked for Remington Rand. Her name was Grace Murray Hopper. She put together a system for the UNIVAC 1 that wasn't as sophisticated as what we were trying to do with FACT, but it was more sophisticated than what Roy Nutt had done with his SAP assembler. It was about a half way step. She and a crew of people pioneered this project. In any of these things there is usually one guy who is the idea guy or pivot person. I'm sure that Nutt had some help with the SAP compiler, the same way we had whole crews putting together the I/O system that General Motors and North American put together. However, there is usually one guy that comes up with the original idea and herds the people around and puts in the key ideas. Grace is given the credit for the Flomatic system. It was a system aimed at commercial data processing, and to my knowledge, it was the first compiler-like system aimed at the commercial field. She gets a first there.

It was being produced about the same time as SURGE, a system which I didn't mention. SURGE was a SHARE system that was produced to do some commercial data processing on the 704. We were all grouping in those days trying to figure out how to handle commercial work. The scientific work was fairly well under control

because it was all algorithmic; we had equations and we had to have compilers, assemblers and systems that would handle equations. We were groping, in some ways we still are, with how to represent commercial data processing transactions and activities. SURGE was one of those early packages put together by the SHARE committee, headed by none other than Fletcher Jones. Flomatic was put together by Grace Hopper for Remington Rand as a piece of company software.

Dick Clippinger at Honeywell was the guy that thought that we ought to have commercial languages that were more like natural languages. He was the guy that set Computer Sciences up in business to do FACT, which was a "natural language" system. It was like English in that it had subjects, verbs, compound verbs and adjectives that modified the breadth of a noun. It was based on good semantic principles. As part of the design of FACT, I somehow learned, or came across, or became aware of a language called Basic English. There is a book entitled <u>Basic English</u>. It was a competitor to Esperanto in the late 1930s and 1940s when the Spanish speaking people thought

PATRICK

Esperanto was going to be the international language of commerce. There were some anglophiles who thought Basic English was a better language of commerce. I went down and got a copy of this book called Basic English, and its vocabulary, as I recollect, contained about six hundred words. I used that vocabulary to choose the verbs, adjectives and nouns for the FACT compiler. We rode on the shoulders of those semanticists twenty years earlier who had developed Basic English as a language of commerce. The FACT compiler was based on some very firm semantic linguistic principles.

I think the author of Basic English is C. K. Ogden. I used that semantic underpinning, because trained as an engineer I knew nothing about language, but I knew that somebody had to have worried about these kinds of things. We used that book to design the FACT system and to structure the way it processed data.

Also about this time there was a group of people that gathered together in Washington, DC to try and define a language for commercial data processing that the Federal Government would sponsor. I've forgotten for the moment who sponsored the committee to begin with, but Charlie Phillips of BEMA was in on it, Grace Hooper was in on it, Dick Clippinger was in on it. Honeywell

PATRICK

decided to cooperate fully and contribute, thus Roy Nutt was a member of the initial group. It was that group that originally defined the COBOL language. We now know it as CODASYL. At that time they were looking for a language which all of the smaller computer manufacturers could implement as a single language rather than building their own. They recognized that going independently into Flomatic 2 and FACT 2 and so on was not a good idea.

At that time IBM had a development group going under Dick Talmadge and they produced a language called COMTRAN, a commercial language also competing with FACT in the same time frame. COMTRAN was done for the IBM 709-90 series by a crew of guys out here in Los Angeles.

Everybody had a competing language and they realized for this commercial stuff that there probably shouldn't be ninety-seven languages, there ought to be one, the same way that FORTRAN was developing into a de facto scientific standard. They tried to create a standard rather than letting one happen and then adopt it, and COBOL was the first effort to do that. It had the full support of the Honeywell Corporation and hence Roy Nutt was a very early member. In fact,

PATRICK

he attended the organizational meetings for all this. At that time we had the users manual for FACT published in first draft form and we were actually ahead of the COBOL people by quite a ways. We were then working on the detailed design to implement it. This is in 1959 before I left Computer Sciences. Those original FACT manuals described how this procedural language for commercial processing was going to process things, and had examples imbedded in the prose of how you would use the language. As I say, the vocabulary was drawn from Basic English and the examples were drawn from that vocabulary.

There are still examples in today's COBOL manual, well that is too strong because I haven't seen one recently, but up until two years ago, there were still examples in the COBOL manual that were lifted from the FACT manual that I wrote while Project Director at Computer Sciences in 1959. Most of the people who have anything to do with COBOL don't know that it is based on firm semantic principles and twenty years worth of linguistic work that I just stole when I picked up that book Basic English.

RM: That's great. I like that. What do you know about JOSS? Did you ever work with JOSS or have any contact?

BP: JOSS was developed at RAND when I was a consultant. The ideas were originally set down by J. Cliff Shaw. JOSS-I was done in the waning days of the JOHNNIAC and the JOSS-II project leader was a fellow by the name of C. L. Baker, called Charlie. Neither one are with RAND anymore. It was a nice system, very user oriented but limited in scope. It was limited in scope for two reasons; one because they were trying to build a little system to do little things, and two it didn't hold files. There was no file handling ability in it, there was no restart built into it, it was for little bitty jobs. For little bitty jobs they didn't think you needed all those things. It was a mistake on their part because little bitty jobs tend to grow big.

You would work on an equation or a few equations in JOSS and then you would have to adopt an entirely different programming techniques and an entirely different computer. Since they didn't provide for a bridge in the JOSS system, it dead ended a lot of applications and hindered the migration and so on.

In addition JOSS-II was done for an orphan machine. It was done for the PDP equipment. The worst thing that could have happened to DEC at that time was that the Air Force had ordered twenty of those things,

because DEC didn't have the manufacturing capability to put them in. It was one of these short sighted decisions brought about by the way the Air Force contracts for R&D. They got the first JOSS machine produced and installed cheaper by using DEC equipment. If you had wanted a dozen of them, the overall cost of ten wouldn't have been cheaper and the overall cost of a hundred would have been impossible because DEC couldn't have built and supported a hundred machines. So it was a one of a kind. Never did go very far.

RM: Tell me a little about DEC. I really don't know too much about them.

BP: Well, DEC started out making digital modules for laboratory work. They built pluggable cards. If you were going to build a computer, or if you were going to build a data reduction system, or if you were going to build one of these tape controlled machine tools we mentioned earlier, you would need some pluggable cards to go into it. DEC built high quality pluggable cards. A couple of guys in the sales department at DEC said, "Say, we could put little boxes around these modules so that they are not just raw modules, we could package them, and if we packaged them we could use them for training purposes." These little packages would sit

on a table and students could learn how to build adders and multiply circuits and so on using these little standard modules.

The next thing that happened to DEC is that somebody was fooling around doing that and he says, "I think I'll take off a month and build a computer with this stuff." They did and the first little DEC computer was built. For many years they didn't know anything about the computer business. They were selling modules. They had built a sales force and a supplier system to sell modules. When we were with Computer Sciences there was a DEC sales office down by the airport. Those guys didn't know anything about the early DEC computers but we could order all the modules we wanted. I actually made a call on them to try to find out what the computer was like and to see if there was a business there. The salesman didn't know anything about the computer. They were selling modules to all the aerospace industry, and their office was right in the middle of the aerospace instrumentation business.

RM: Do you know approximately when DEC was started and who were some of the key people?

BP: I'm now reading from the 1972 Datamation Industry Directory. "The Digital Equipment Corporation is in Maynard, Mass. and it was established in 1957. As of 1972 it had over one thousand employees and less than ten thousand employees, and fifty million dollars in sales." They have offices everywhere and a whole string of products.

RM: Do we know some of the early principals?

BP: No, I'm afraid I don't know any of the principals of DEC. They were at the other end of the world from me in Boston and I was out in California about that time.

RM: Okay. In Number III of your outline, which is software technology, did we get into table driven processes?

BP: No, we didn't touch on that.

RM: Maybe we should?

BP: Yes, I think we should. Let me see what I can do as far as memory goes there. If I had an application package for banking, or for accounting, or for a payroll or personnel application, and I wanted to make it general purpose, I would do that by writing table driven code, or writing a processor that processed table driven code. That was one of the landmarks in software technology. It is obvious now. It is in

all the textbooks and taught to every eighteen year old kid that goes to college and takes a first course in computers. However, that was a pretty major thing when it was done. I'm going to have to search around in my mind for awhile to find the name of the guy that did that. There were some people like Tom Steele of SDC, Erwin Book of SDC, and Jules Schwartz of SDC who were probing around in the right place at the right time. There were guys like Bob Barton of Burroughs, now up at the University of Utah, that were probing around in these areas, obviously IBM had a whole crew of, at that time, nameless designers plus the people we've mentioned like Backus and Talmadge and so on. But I don't think any of those guys did it. It was a military. It was a contractor up in the Boston area doing mostly military work. I think the name was Tech Ops.

I have before me a compiler manual for the AIMACO compiler, dated 30 June, 1959. It's an Air Material Command manual. AIMACO is a cousin to Flomatic, both of them done by the same Remington Rand crew as I recollect, and Grace Murray Hopper was in the middle of both of them. AIMACO was for a UNIVAC 1105 system. I'm sorry to say my memory fails me and I don't recollect how it relates to FLOMATIC, which was first and which

was a derivant of the other. Here, at least, is an AIMACO manual that has surfaced from the bowels of my garage.

RM: I think we should talk about the RAND approximations.

BP: I also have in front of me a specially bound copy of the RAND Approximation. Inside the RAND building these were called Form 15s. They were prepared by Cecil Hastings. He made them so that we could do computing in a practical way. They were first used, I believe, for the 604s (in card form), then the CPCs, and they formed the basis of much of our work in the early 701 days. Cecil was a mathematician who believed that computing gave numbers, and it was important to get numbers properly. He prepared these approximations so that we could find logs, square roots, sines and cosines and arc tangents when we needed them, using the early primitive computers. This particular set was sent to me when I was still at Edwards Air Force Base. It was transmitted from RAND to me at Edwards and that puts it in the 1952 early 1953 time frame.

The third document I have in front of me is a master document called the GMR-NAA I/O System. In November of 1955 I prepared a presentation for the third meeting of SHARE which was held in Boston. The first twenty pages or so of this Acco binder is the presentation and a handout I made at SHARE. The last hundred pages

or so of this binder are the details of the General Motors half of the joint North American-General Motors project. I was the project leader and this is a binder full of my project leader's notes that I just collected and kept together as the project progressed. At the tail end of this manual you'll find several sheets shoved in the back which are memos from me to George Ryckman. George Ryckman was the General Motors manager who completed that I/O System. The design was almost all done and was well into the implementation stage when I left the project to go to work on the THOR guided missile application. You will find a few notes written by me and addressed to George Ryckman dated June, 1956. I was clarifying some of the design intent so Ryckman and his crew could complete the implementation of the I/O System. So much for the major milestones we found in my garage.

There are major milestones in most every facet of computer technology. We talk about the relay, the tube and the transistor being milestones in the hardware area. We can talk about interpretors and early monitor systems and compilers as being milestones in the monitor and software area. It seems that for

the record there are several major application milestones that we ought to consider recording for posterity. Most of us in the computer field plodded our way forward, doing applications of a moderate size and while the mathematics or the application we were attempting to solve might have been new or unique, there wasn't anything stupendous about the undertaking itself. There are several of these, however, that were noteworthy. In the very early days, GE attempted to automate all the information in their Louisville factories. John Swearingen was part of the project. I think as we look back on it, it was a most audacious undertaking for its time which was about the mid 1950s, I guess. It was only partially successful. They spent an enormous amount of money and had a hell of a lot of trouble. They attempted to automate all of the management information system for a whole factory. While they failed in some senses of the word, their undertaking was in fact a success and they learned quite a bit about what to do and what not to do.

RM: Was this a UNIVAC I project?

BP: Yes, I believe it was on the UNIVAC I.

Another major application which kind of stood out from the rest of the things was the SAGE undertaking. This was an outgrowth of some early work

done by Lincoln Labs at MIT. They ran a pilot control center somewhere out on Nantucket Island, and then they decided that they wanted to put a net of these monster computers clear across the United States connected online in real time to radars. It was a monster set of computer programs. It was the biggest computer system that had ever been considered, much less undertaken in that time frame. And they did it. It cost a whale of a lot of money and a lot of guys worked like hell at it.

Wes Mehlan of SDC was a focal point in that early arena. He later went on to be president of SDC. A person who could give more information on that is Pat Haverty who now works for SDC. He worked for RAND and then he went to work for SDC and then back to RAND and now he is back at SDC again. Pat was part of some of that early work in the planning of setting up this interconnected net of radars and computers to guard the northern boundaries of the United States from potential manned aircraft attack.

The next activity that was a major step was the data reduction work which was done in many engineering shops. There were many flight tests run; there were

many wind tunnels being run. All of this data was fairly well behaved, but there was quite a lot of mathematics to convert it to a form that could be plotted and compared to allow you to evaluate the engineering design. While it wasn't one huge application in the sense of SAGE, it was like a ground swell that passed across the United States. All of a sudden every engineering organization was doing data reduction in a large volume sense. We just did a lot of data reduction in a very short time. Then, of course, it became common place.

About the time we were doing that, the commercial side of the field undertook to do a lot of accounting. They started to do accounting; simple bookkeeping, replacement of clerks. Then the magnetic tapes in the computers started to become reliable enough to hold data files. All of a sudden another wave swept across the country and instead of doing trivial things of a bookkeeping nature, we started doing larger and larger accounting activities.

About the time the data reduction and accounting hit production status and became somewhat mature, engineering design was mature enough to be the next wave that came in from the technology. The mathematicians

PATRICK

knew enough about the computing to do mathematical approximations for the computers; the engineers had started thinking in terms of equations rather than taking blocks of metal and files and hogging out something to see if it flew; and we started doing engineering design mathematically with computers to predict the performance of things before they were built, to do the stress evaluations with computers rather than desk calculators or mechanical hand calculators. All of a sudden we were doing bits and pieces of engineering design. As we did more and more of the engineering design, it became clear that we could do entire engineering designs.

That moved us into where we started modeling physical processes. We built mathematical models, you can call them simulators if that makes you feel better. These were large complexes of applications, so that if I had engines and transmissions and gear trains there were three computer programs one could feed data into the next one. Pretty soon I could model a whole power train from the time the engine was started to the time the automobile was moving smoothly down the highway.

Or, if I had manned aircraft, I had separate programs for the control surfaces, the engines, and the aerodynamic surfaces, and all of a sudden I could

PATRICK

hook these all three together and I could model a manned aircraft in maneuvering flight.

When we started being able to hook these individual computer programs together, there were no trumpets blaring. It was a rather major breakthrough to be able to do a real thing in its entirety. About 1954 or so, we were able to put these engineering designs together and make models of them as I indicated.

Shortly thereafter a fellow by the name of Jerry Haddad in IBM was in charge of the Advanced Systems Development Division. ASDD looked around for places to apply IBM computers and they pounced upon the airline reservation system as being a natural for computer processing. They mocked up some terminals and consoles and started working with American Airlines. I've forgotten the total development time, but it was six or eight years, and it finally resulted in the SABRE System which today, in 1973, has two thousand or so terminals stretched out all over the United States, several computers in Briarcliffe Manor, New York and a monster data base. If we hadn't done that with the computers, the airlines would not have been able to cope with the passenger loads that came along when the first jet aircraft, the 707s and DC-8s began to carry a hundred or so passengers at a time. The airlines

PATRICK

couldn't have handled the reservation load. So here was a case where the fallout from the military technology provided the engines and the air frames to provide quick mass transportation, but we couldn't have scheduled them, we couldn't have done the flight profiles on them, we couldn't have sold the seats and made it profitable if it hadn't been for the computer technology coming along in parallel.

Shortly after that took place, I guess it was 1962 or thereabouts when these things started to come together--I wasn't in on SABRE so I'm not sure of the dates--we started doing some serious process control work where the computer actually became a dynamic member of the system producing a product. Maybe the computer was part of controlling a series of pumping stations, but here were cases where the computer finally got reliable enough that man would trust it with doing some real control functions where there was a danger of loss of life or high economic risk. When the computer became of age, if you will, in the process control area, we could then do air traffic control aided by the computer, we could then put computers into the process industries like Du Pont did, and like Texaco did in their refineries. The computer then allowed one man or a few men at a central site

to control a much bigger comples of petro-chemical equipment. The larger complex, in many ways, was much more efficient, and with the efficiency the prices came down.

There may be some other milestones I've overlooked, but those seem like the giant strides that were made in the applications area.

RM: Who were some of the people who were doing the early work on process control? Can you recall?

BP: As a matter of fact, I don't know. I wasn't part of any of the petro-chemical applications at that time. While they were doing that, I was working at General Motors. We were working on modeling and simulation of physical processes, and I wasn't in on any of those other developments so I can't supply you with any names in that area.

RM: During this period of frantic and great activity, what was the relationship between the hardware people and the programming and software people? Was there interrelationship? Was there communication? Was the one attempting to assist the other?

BP: Well, if I go back far enough, back to 1952, the hardware people and the software people were totally divorced, I don't think they even knew that each other existed. When we got the 704 machines, the hardware came out

PATRICK

essentially bare bones. The SHARE work and the SHARE Operating System said that the customers were willing to bank together to produce the software, and that was for the 709. The whole plan was that the SHARE committee would design the thing and IBM would produce it. That, in fact, did happen; IBM did implement the SOS system, or chunks of it, and the design was done by user volunteers. With that the users gradually got the IBM Corporation to consider, believe and understand that the machines had to have software with them or the customers weren't very interested in buying them. This thought wasn't, to my knowledge, invented by the IBM Corporation. They were willing to sell bare bones hardware, but the users said we need operating systems, we need compilers, and we need utility programs to use them. So IBM said, "Say, if that is a general need of all of our users, maybe we can up our price a little bit, supply software, and sell more machines." It wasn't an IBM invention; it was a user innovation.

RM: It was a response then to the need of the user.

BP: Yes, ma'am. We've got to go clear up to 1959-1960 I believe to the Burroughs B-5000 which was done by Lonergan, Barton, and crew. I believe that was the first machine where the hardware people and the software

PATRICK

people actually joined together to put out a computer. IBM machines up to that time, and even considerably after, were designed by engineers and then programmers built the software. Two clear-cut sequential steps. I was working as a consultant on the B-5000 project with Barton and all, and there was an attempt to get a clear symbiosis, a working together, a harmony, between the hardware side and the software side. I believe the spark plug of that is Bob Barton; he gets credit for that. I think that was the first attempt to put those together that way. Once they did it things started happening, and it wasn't much later that everybody started getting hardware engineers and software programmers working together.

At one time, it was in the early 1960s, IBM had alternate layers of their development organization intermeshed with engineers and programmers as a way to force the melding of these two disciplines. A programmer was bossed by an engineer who was bossed by a programmer who was bossed by an engineer. Of course IBM now has a whole army of people who are cross-trained in both fields. To my knowledge, there has been little applications intercourse even today. There is a fine harmony in many companies between the hardward and the software, but as far as the

PATRICK

applications go, there hasn't been all that much effort to determine the application's needs before laying down a design. As far as the computer center manager goes, he is one of our unsung heroes of 1973. A lot of our big shops are not being operated well because they weren't designed well. They weren't designed with operations in mind. They just happened and they are very difficult to manage. We still have a ways to go.

RM: Bob, thank you very much.

END OF INTERVIEW

Index - Robert Patrick
26 Feb 1973

Index - Robert Patrick
26 Feb 1973

Index - Robert Patrick
26 Feb 1973