Oral History of Captain Grace Hopper

Interviewed by:
Angeline Pantages

Recorded: December, 1980
Naval Data Automation Command, Maryland

CHM Reference number: X5142.2009

Table of Contents

Captain Grace Hopper

Contributed to the Software Industry SIG – Oral History Project

Abstract: In this 1980 interview, Grace Murray Hopper describes her entry into computing and programming, when, as a Navy officer, she was assigned to work with Howard Aiken on the Mark series of computers and its coding. She talks about her post-Navy years with John Mauchly at Eckert-Mauchly Computer Corporation and, subsequently the Univac division of Sperry Rand where she and her staff developed A-O, the first compiler, followed by the first English-language compiler, B-O, or Flow-Matic, the major predecessor of COBOL. And, ultimately, about her return to the Navy to work on standard languages and software, during which she continued her earlier work with CODASYL to create COBOL, i.e., business-language standards.

Her discussion is filled with the technological, industry, and human issues she considered critical: Small systems and distributed computing in place of giant general purpose systems. The full integration of computing into the organization; Teaching the young and encouraging them to use their unfettered imaginations and pursue new ideas; Common sense and communication; and the honed skill of writing and documentation.

***Note**: The following interview with then-Captain Grace Murray Hopper was conducted in December 1980 by Angeline Pantages, former international and senior editor for Datamation magazine. This interview, as well as interviews of Hopper's colleagues from various eras of her work, was for a biographical series, "They Made the Future in the Past," which appeared in Denmark's computer publication, Data. The article on Grace Murray Hopper was published in February 1981.*

Background History

To clarify the discussion and the timeline of events in her life, Ms. Pantages has provided this brief history.

1906-1943

Grace Hopper was born in New York City December 9, 1906, the eldest of three children. Attending Vassar College, she graduated Phi Beta Kappa with a BA in Mathematics and Physics in1928. She earned an MA in mathematics at Yale University in 1930 and a PhD in mathematics in 1934. In 1931, she joined the Vassar faculty as associate professor – just a *year after she married Vincent Foster Hopper, a doctor of comparative literature and instructor at New York University.*

Twelve years later, Dr. Hopper decided it was time to serve her country, and – as it turned out by chance – to begin her illustrious career in computing.

1943-1949

In December 1943, Dr. Grace Murray Hopper was sworn into the U.S. Navy Reserve, and, in 1944, was commissioned Lieutenant (JG) and ordered to the Bureau of Ordinance Computation Laboratory at Harvard University. There she was assigned to program (actually to code) one of the first, if not the first, program-controlled computers ever built: the Navy's electromechanical Automatic Sequence Controlled Calculator. The behemoth (55 feet long, 8 feet high) is known to the world as the Mark I computer. The chief designer and head of the project: Commander Howard H. Aiken.

In 1946, released from active duty, Dr. Hopper joined the Harvard faculty as a research fellow in Engineering Sciences and Applied Physics at the Computation Laboratory to continue work with Aiken on the Mark II (a paper-tape sequenced calculator) and Mark III (an electronic computer with magnetic drum storage).

1949-1964

In 1949, her three-year fellowship at Harvard ending, she decided to join the Eckert-Mauchly Computer Corporation to help design and program UNIVAC I, the first commercial electronic computer. This pioneering company was sold to Remington Rand in 1950, which in 1955 merged with Sperry Corp. to create Sperry Rand.

In 1952, Dr. Hopper developed the first compiler, A-O, a mathematically oriented single-pass compiler. As Director of Automatic Programming Development for the Univac division of Sperry Rand, her A-O work subsequently led her and her staff to create an English-language compiler, B-O, known later as Flow-Matic, a precursor to COBOL. In 1959, Dr. Hopper and colleagues among users, educators, and vendors began their push for a common language for business applications and hence, greater compatibility among vendor systems. They organized CODASYL, the Conference on Data Systems Languages, which created COBOL and advanced its development over the years.

In the 1950s, Dr. Hopper, impatient with the growing complexity of the all-purpose giant mainframes and huge operating systems, also began advocating the use of smaller computers and distributed computing.

***1964-1967.** When her programming development group was disbanded at Univac, Dr. Hopper – not wanting to move up in line management – moved with a small staff to the University of Pennsylvania Moore School of Engineering. As she had done as a visiting lecturer and professor at Moore School over the Univac years, she taught Penn students and lectured around the country on the future of computing.*

***1967-1986.** While she retired from the Naval Reserve with the rank of Commander in 1966, Commander Hopper was recalled to active duty in 1967 for a six-month assignment requested by Norman Ream, Special Assistant to the Secretary of the Navy for Automatic Data Processing. Ream had the length of her assignment changed to "indefinitely" – this, because he wanted her to execute her proposal: the development of testing and validation procedures to enforce standards and hence, ensure portability. (At this point, vendors had so hampered portability among systems that conversion costs were exploding.)*

From 1967 to 1976, she served as director of Navy programming languages and language standards, and from 1976 to 1986 she was Special Advisor to the Commander of the Naval Data Automation Command, heading the Training and Technology Directorate.

She was elevated to Captain in 1973, rare for Navy reservists, and a decade later, in 1983, she was promoted to Commodore. The ultimate Navy honor came in 1985, when she was promoted to Rear Admiral, an historical first for a woman. In 1986, Rear Admiral Hopper, the oldest serving officer, retired and was honored in a ceremony on the USS Constitution.

***1992.** Rear Admiral Grace Murray Hopper left us January 1, 1992 and was laid to rest at Arlington National Cemetery. Her life, spanning almost nine decades, was filled with historic accomplishments, innumerable friends and colleagues, and more than 30 awards from the U.S. government and professional societies and educational institutions around the world. But, as her interview illustrates, she was just as – perhaps most – proud of working with the young, teaching them about the future of computing, and encouraging new ideas and how to communicate them. She was and is accurately dubbed by all, Amazing Grace.*

[This transcription begins with Grace Murray Hopper's time at Harvard, working with then-Navy lieutenant Howard H. Aiken on the Mark I, II and III calculators. The tape, as transcribed, begins in mid-sentence.]

1943-1949: Mark I, II, and III Computers at Harvard Error! Bookmark not defined.

Grace Hopper: [I started to work on the] Mark I, second of July 1944. There was no so such thing as a programmer at that point. We had a code book for the machine and that was all. It listed the codes and what they did, and we had to work out all the beginning of programming…and writing programs and all the rest of it.

We stayed there through the building of Mark II and Mark III. What was carried over from Mark I was that, after a while, each of us began to build a notebook, what turned out to be subroutines. We didn't know they were; we called them programs.

If I needed a sine routine, angles less than five over 4, I'd call over Dick Bloch* and say "Can I copy your sine routine?" We'd copy pieces of coding.

Angeline Pantages: What was your function with the Mark series?

Hopper: I was a mathematical officer. We did coding, we ran the computer, we did everything. We were coders. I wrote [programs for] both Mark I and Mark II. Commander [Howard H.] Aiken came up to my desk one day and he said, "You're going to write a book." I said, "I can't write a book." And he said, "You're in the Navy now." And that settled that, and I learned to write a book.

Pantages: When you look back at that period, did you have a sense of history or was it an evolution?

Hopper: It was a steady evolution to doing things easier and better, more correctly. Dick Bloch was the only person I ever knew who could write a program in ink correctly the first time. He could, we did. And, I guess during those years, up as far as 1946, we didn't think ahead. The whole thing was the war, the end of the war, getting a job done, terrific pressure. We didn't think ahead at all.

It wasn't until after that that we ran the first problem for Prudential. This was when we used two digits to indicate alphabetic characters on Mark I. They (Prudential Insurance) were the first ones to try to do anything in data processing on a computer. . We ran a program for them – I've forgotten what it was about – but we used two digits to show each alpha character. Mark I was only numeric, it had no alpha. That opened up a new perspective that none of us every thought of. It was totally new.

[U.S. Navy ensigns Richard Bloch and Robert Campbell began working on Mark I in 1943, while it was still being assembled at IBM, and were involved in programming, development, and maintenance at the laboratory for the next few years. Both moved on to Raytheon. Bloch later

went to GE to develop large systems, while Campbell went to Burroughs, as a systems analyst, and to Mitre Corp.]

[*Further discussion on this era begins on page 21.*]

1949-1964: Eckert and Mauchly, Univac, and the One-Pass Compiler

Hopper: Then in 1949, I was at the end of my time at Harvard, and I joined Eckert and Mauchly *[at the Eckert- Mauchly Computer Corp.*]**.** John Mauchly hired me. Again, we built up those pieces of coding.

It was not until 1951 when I was given the assignment to collect them all together so that we would have a library of subroutines that we could all use, and some means of getting at it. Now by then we had the book from Wheeler, Wilkes, Gill, and they had their whole family of subroutines for the EDSAC, remember?

[*Maurice V. Wilkes, David J. Wheeler, and Stanley Gill – professors at the University of Cambridge and builders of EDSAC, the first storage program computer – authored "The Preparation of Programs for an Electronic Digital Computer," in 1951. This is the first textbook written on computers, programming and software.*]

As far as I was concerned there were two problems. One was you had to copy these darn things, and I sure found out fast that programmers can not copy things correctly. On Univac I we used a "delta" for a space, and if he wrote a careless delta it could look like a 4. We also put a decimal in Univac I. Any number of people used Bs that turned out 13s. Programmers couldn't copy things correctly.

On each of these routines they started with zero, which when you put them into a new program you had to add every one of the addresses to position it in the new program. Programmers could not add.

There sat that beautiful big machine whose sole job was to copy things and do addition. Why not make the computer do it? That's why I sat down and wrote the first compiler. It was very stupid. What I did was watch myself put together a program and make the computer do what I did.

And that's why it was a single-pass compiler and that always astonishes everybody. Because obviously you couldn't do that, because you'd run into a place where in the flowchart, for instance, you could have two kinds of tests. One would jump back to somewhere you'd been and you knew where you put that program piece, but you would have another test that jumped forward and you haven't processed that yet, so you didn't know where it was going to go.

So I developed what I called neutral corner – I got it from playing basketball. This was to go forward, so I had a jump down to what I called neutral corner. Then I put a flag up, I've got a message for you, and in each step, each glob, I looked down to see if I had a flag for it. And if I did, I'd put another jump from there to that operation.

At the cost of one extra jump I had a one-pass compiler. You went zooming through the computer, you didn't go back, you didn't have separate runs, and all that stuff.

Pantages: How did you get together with Eckert and Mauchly?

Hopper: In 1949 when people knew I had run out the time at Harvard, (and I guess everyone in the industry knew it) practically everyone asked me to come for interviews, including IBM. I went to the IBM headquarters and they gave me a huge *[offer].*

I was one of the very few people who did not work for IBM. I went for interviews with practically every computer manufacturer that there was at the time. Honeywell, RCA was thinking about it, Burroughs was in it.

But it was John Mauchly I just couldn't miss. Working for him was obviously going to be a great pleasure. He was a wonderful guy, one of the best that ever lived.

Pantages: Did they want you to do something special?

Hopper: No, they were then building the programming group for Univac I. At the interview there was both John and Betty Holberton. And Betty has never received the credit for the work she did and should have received long since.

Pantages: Where is she now?

Hopper: She's retired. I think she's still part time at the National Bureau of Standards. Everybody's forgotten that she wrote the first program that wrote a program. She wrote that sort-merge generator, and what she did was feed in the specs for the data you were handling and the keys and that sort of thing, and then it generated the sort program for that specific data. That's the first time to my knowledge that anyone used the computer to write a program. Betty did that. I don't think she's ever fully received the credit for what she did in that case.

Then she got out and got buried in NBS. Nobody heard from her after that, which is just a shame. That first work she did was that automatic generation stuff for Univac I. I'm not sure that I would necessarily have gotten done what I did get done if she hadn't been ahead of me, so to speak.

Knowing that she had used a program to generate a program, I had a good deal more nerve to go ahead and build the first A-O compiler. It finally got called AT-3.

That's the one where I got into trouble. We built a preliminary compiler, a model which would only take 20 statements but had all the controls in it, in order to get the budget. We had this program, but the more we looked, the smaller it looked. It looked so trivial and we were asking for the biggest budget we had ever asked for. So we decided we ought to do something more.

We had carefully fixed it so that each statement began with a verb. Now the only type of statement that is coming across in all languages is the command imperative. We simply substituted the words in French and German and ran the program in French and German, simply the verbs and nouns. It showed that we could do it not only in English. But that idea got clobbered because it was perfectly obvious that a computer built in Philadelphia could not understand French and German.

But that was in the original program. Eventually it did appear in most languages. And then what happened was even stranger. You look at programs in Sweden and you find English verbs and connecting words and nouns in Swedish. In Germany, you find English verbs and connecting words and nouns in German. In Japan, it's even more so, because, if you look at the Japanese COBOL text, you find verbs and connecting words and Roman characters in English and the nouns in Japanese and Japanese characters.

Actually, COBOL has become a language you can talk to programmers around the world. We had proposed that originally and it sure got clobbered.

Pantages: I guess I don't understand…why would they clobber it?

Hopper: It was obvious that an American computer in Philadelphia couldn't understand German.

Pantages: Who was making these obvious statements?

Hopper: Non-computerized management; Marketing particularly.

Pantages: Univac would have cornered the international market.

Hopper: I would have thought it would be useful to NATO, because I thought they had the common verbs for the things they were going to do. And the nouns, they'd just have to have a dictionary for things they were referring to for inventory control and other things like that. They'd have common nouns throughout NATO, and they could make a dictionary of common verbs and translate the program. You could write one in English and you could translate it and it could go to the xxxx. No problem, you'd have communication. It would be a limited vocabulary.

The Need for User-Friendly Languages Error! Bookmark not defined.

Pantages: At that point did you have a feeling for what was happening, in terms of what you were contributing?

Hopper: No. I've always objected to doing anything over again if I had already done it once. That was building the compiler. Then I decided there were two kinds of people in the world who were trying to use these things. One was people who liked using symbols – mathematicians and people like that. There was another bunch of people who were in data processing who hated symbols, and wanted words, word-oriented people very definitely. And that was the reason I thought we needed two languages.

The data processors did not like symbols, abbreviations that didn't convey anything to them. They were totally accustomed to writing things in words. So why not give them a word-oriented language? And that was part of what was behind Flow-Matic B-0, which became one of the ancestors of COBOL.

And that still makes sense to me. The other thing that bothers me is why we aren't developing more languages. You'll notice that when they began to program machine tools they got together all the people that used machine tools and developed the F APT language for programming machine tools.

I still feel we need a lot more languages, not one. Everyone always wants to make the one all-purpose do-everything language. I can make that very easily. All we would have to do is put every bit of English and every bit of mathematics into the computer and there it would be. It would be too big for everybody, too cumbersome. We would be continually trying to design subsets.

What they should have done was build more languages for particular application areas. Now for instance, one of the ones that was most needed was biomedical for all these things in [various] units and the experimental work in biology. Why don't we have a biomedical language that includes the words, the verbs, and the mathematical techniques that people in the biomedical field use?

Pantages: It wasn't MUMPS… *[MUMPS is a programming language created in the late 1960s, originally for use in the healthcare industry.]*

Hopper: It’s an English language, but it’s interactive, and of all the information you have to protect, it’s medical information. Hospitals are using it and because it’s interactive, you can get in there with no trouble at all. They haven’t discovered that yet, and it’s not going to protect privacy. But don’t put that in the article please. Let them find it out themselves. They’ll have to sooner or later.

Pantages: It sounds like an issue we should all be talking about.

Hopper: Well, we’ll get around to that later. It’s a very powerful group out there, the National Bureau of Standards, that is pushing MUMPS and I’m not about to fight [them]. Okay.

What I was after in beginning English language was to bring another whole group of people able to use the computer easily. The people that used English, not symbols. It w as my own understanding of the different kinds of people, and I knew darn well most people didn’t like symbols. And you had to give them something else to work with, and obviously the next-best thing was English.

Pantages: Again, did you realize the impact of your work?

Hopper: I kept calling for more user friendly languages. I’ve always tried to do that, that’s why I want these other languages that are aimed at people. Most of the stuff we get from academicians, computer science people, is in no way adapted to people.

[*At this point, Hopper was handed a brochure about skiing in Bangor, ME, and her eyes lit up. ”Oh boy look at that… Bangor, I love Bangor. I used to like cross-country skiing very much. I know I wasn’t going to get a chance to do that. And besides that, my bones were brittle enough by then so if I fell down I’d break something.”*]

Now that you’ve made me think about it, I think I’ve always been driving to make those things easier to use by everybody. Feeling that we needed greater use of it, more information processing, something had to be done to get people to use it.

Pantages: That was the driving force. It was kind of funny given your background and your personality, having come out of the sciences, mathematics.

Hopper: I wanted to be an engineer, but there was no place for women in engineering when I graduated.

Pantages: I was going to ask you what you wanted to be…

Hopper: That was 1928, you see. My grandfather had been a civil engineer, and he was a senior civil engineer in the city of New York. He used to take me with him when he went out surveying, and he let me hold the red and white pole. And he also let me look through his gadget. And I wanted to be an engineer.

My dad always made things, and I've always been fascinated with how things work and making things work. But there was no place at all for women in engineering when I graduated in 1928, and that's why I went to the Math-Physics Center because at least they were making things work.

I've also for some reason been able to explain things to people without necessarily using a technical vocabulary.

Pantages: I noticed that about some of your writings in the 1950s – that I could understand them.

Hopper: I could switch my vocabulary and speak highly technical for the programmers, and then tell the same things to the managers a few hours later but with a totally different vocabulary. So I guess I've always been innately a teacher.

So that again was what made me want to get user-friendly languages out so people could use them.

[*Further discussion on this era begins on page 25.*]

1980: Demands for the Future

Application Processors, Database Machines, Distributed Processing

Hopper: I could see people were going to need these things and the amount of information would increase. And I still think it's going to increase even more. I don't think we've even begun to recognize how much we are going to have to do with these computers. I don't think people are facing what the future is going to be like. The big computers are not going to handle all the data. We'll go to systems of computers. And the sooner we begin to do it the better. There's never any reason to put inventory and payroll on the same computer, but we only did it because we had only one computer. There's every reason to pull those off onto separate computers and dedicate them. We can simplify the software and run them in parallel, which makes things go much better.

It will take a while to persuade people.

Pantages: They are preaching it. They are talking about applications processors.

Hopper: They are not there yet. I want everyone that has the big computer to gradually pull everything off and put it on small computers. And one of the biggest assists is going to be this database machine that just appeared. One was shown at NCC [National Computer Conference] and ADABAS had a whole article about it in *Computerworld*. [*ADABAS, a Software AG product, was one of the earliest commercially available database products, first released in 1970.]*

Pantages: Oh, the one they are doing with Magnuson.

Hopper: There's another one. I think we'll hear something from Cullinane before long, I have a suspicion. But the whole point is you get an independent database on one, two or three computers and all the other computers can access it; there's no longer a need for the big central processor. So the concept of the database machine is just tremendous. You know that came out in 1974 at Bell Labs – they had the paper in *ACM Communications* and they showed it cost less and went faster. It's taken six years to get it across.

Pantages: Well there isn't much new out there that wasn't discussed years ago.

Hopper: It takes five to 10 years for a new idea to get across and get used. It seems so anyway. It did for the compilers, it did for COBOL... I see that in the future everyone isn't going to have one big computer; they'll have a whole mess of computers dedicated to specific functions. This will increase the speed of reaction on any file or any bunch of data, because you can specialize the software for that particular job. So it always goes faster. Speed is going to be the pressure. Storage you can buy. It's trivial these days. The speed of response is why the [new communications system] is so terrific.

Pantages: Which one, the distributed network…?

Hopper: Yes. Travelers [Insurance] is going in that direction.

Pantages: Citibank has had its starts and fits.

Hopper: Chase is way ahead of Citibank.

Pantages: How will the distribution and miniaturization of systems impact management and people?

Hopper: Well, in the large companies I would expect that the thing would be managed just the same as at Southern Railway. Jack Jones is the vice president of information; he has all the money. And he's pushing them out to agents and stuff like that. Celanese is doing the same thing, Mike Samek.

You'll find them pushing them out to users and communicating with big databases and so on and so forth. It's beginning to happen at almost all the big companies.

Demand for Programmers and System Analysts

The other big step that's happening is with small users. And I'm finding as I go around the country, I'm finding a tremendous number of small local software houses, who would go with the small businessmen and help them select a computer. Help them get their programs written. And the small user is beginning to use high school graduates, not college graduates, because they can't afford college graduate programmers.

We've got to push computers into schools. They should be in every school, so kids can grow up with them. A lot of them don't have them and don't have counsel. If you watched a four-year-old with Little Professor, you can see a bunch of four-year-olds learn arithmetic. If you give a six-year-old Speak and Spell, you're going to get a generation that can spell again.

They have no hesitation whatsoever using a calculator. They step right into a computer. You give them a computer to play games with and they get tired of it and pretty soon they're programming it to do everything under the sun. They also are not afraid of making mistakes, they have no prestige to live on, and they have no fear of failure yet. That goes up to 17-year-olds I get in here. They can afford to try anything. That's why they are moving so fast. There's no mystique.

For a long time we're going to have an acute shortage of systems people, who can look at a tremendous information system and identify the subsystems and sub-subsystems and the interfaces between them.

We need more of that systems engineering analysis training. The same kind of engineering they did on a missile. They identified the nose cone and the payload and the guidance and the motor. They described each of those elements and interfaces between them. They were farmed out separately.

We haven't taken that systems design now, because we are going to have to look at the total flow of information joining an organization. And implement that information. You are not going to buy a computer for program time. They are going to look at the information flow first, and then they are going to select the computers.

The Value and Cost of Information

I’ve also been having some arguments about what is the value of information in processing. I made up an extreme example in a chemical plant totally operated by computer. Two pieces of information come in simultaneously and one comes from a valve in the plant and says if you don’t open it, the plant’s going to blow up. You have less than a minute to act, 100 lives and $100 million plant. In the same instant comes information that Joe did two hours of overtime. Which is the most valuable piece of information and what are the criteria?

Pantages: Now that one’s obvious.

Hopper: It should be within an organization. I found out how priorities are assigned today: to the senior squeaky wheel, not to the most valuable information. And you can go on two curves. *[Drawing*.] Dollar’s here. Events here. Value of the information goes up quite sharply immediately after the event. But over time, it levels off, and eventually, it’s either replaced by a new piece of information, or you send it to the historical files and keep it for the IRS. So the value becomes something like that [diminishing line].

What about the cost of that information? It’s very low at the time of the event. The further you get away from the event, the more it costs to get more information, the more it costs to store and maintain it. There’s a crossover point there. That’s the point at which we should be getting it out of our on-line files and we are not doing it. Because we don’t know our value and our cost.

And I am asking people to find out the value and cost of their information. Because all we are doing now is leaving stuff in the on-line files. And they are mushrooming. The more you leave it there, the more you slow it down. But the cost of those two curves when you get it out – no one is even looking at that.

Pantages: Their vision is limited….

Hopper: Many people think they have virtual storage, but it was nothing but overlay.

The Navy’s Dilemma: Micros and Software Creation

[*At this point, Richard Burdette, a Naval officer and in management within the Naval Data Automation Command, enters her office. and they discuss the problems being created by personnel who are acquiring their own computers and creating programs they need.*]

Hopper: Hey Burdette, come here and meet a friend of mine. You must have heard of Dick Burdette. He’s got some good ideas too. We’ve been working together since I came back from active duty in 1967.

Richard Burdette: Hardware is cheaper than ever before, but software isn't any cheaper. It's getting very expensive, it's labor intensive, and you have the problem of people reinventing the wheel.

Hopper: But most of the stuff is Wang's… they are using the software that comes with it. Wang's are the friendliest things you've ever seen.

Burdette: You are thinking of word processors. I'm thinking of the micros and minis, where people are innovating and doing their own things. And a lot of people are doing their own things which are the same as everyone else's own things. That's the problem we're having in the Navy that we are concerned with.

Hopper: Most of them aren't spending much for the software.

Burdette: It's now almost invisible as to what they are spending because the user himself is involved. You don't call these people programmers anymore, but they are spending time generating programs. That might be good because now the user can be responsible for his own product.

Hopper: I think that is good.

Burdette: He knows what his requirements are. If he is educated enough to use a computer, we might get programmers replaced?

Hopper: Making it easier for people to get training. This – they are pushing – not to limit the use of computers, but to get people into here and let them find out what they should require and what they should know, etc. and then let them out on their own.

Burdette: What do you think about what's happening at Norfolk with the micro group?

Hopper: What are they doing?

Burdette: We set up a Center of Excellence for micros, for people to become extremely knowledgeable in the technology. Act as consultants to the rest of the Navy, those people who fall below the $10,000 threshold, who can go out and buy computers without permission.

Hopper: She wants to put a $2,500 limit.

Burdette: Ah so. The problem there is that none of these computers – if people did this totally on their own – would be able to talk to each other. There is no standardization. The concept of the Center of Excellence – the group that we hope will be consulting to those who don't know much about computers – will provide an ad hoc type of standardization. Will give them general guidance, so they can within a certain framework go out and get a certain kind of machine that will talk to each other: Standard microprocessors that will handle the same software, the same operating systems. And then we can share programs. If programs are developed in one place or another place, one ship another ship, we can start to throw these programs in a big library through the Center of Excellence hopefully. Then they will get to know what's available and people can come in and say, "Hey I want to do something but before I go ahead and do it, do you have something?" We would be able to say yes we do, we have systems-compatible machines. That's a de facto standard at this point. But it's not a real standard because we have allowed these people to go out, under that threshold, carte blanche, and get what they want.

I guess that's why this gal from DOD is trying to put the clamps on it. Is that Scarlet Curry?

Hopper: That's the one. I'm trying to get her to soften it up. She's writing it.

Pantages: How do you solve that programming problem?

Hopper: Most of them are doing something local, quick and dirty. If they aren't going to use it anywhere else, you don't need it anywhere else.

Pantages: Isn't that a definition that has to be made before anyone gets uptight about it – what information you don't care about?

Hopper: It's a problem when you try to go in with an overall thing. The Philadelphia Naval shipyard and San Diego shipyard were just totally different, because they were working on different types of ships. The Newport one was even more different because they were the only ones that had submarines to work with. So people were different, parts were different, everything was different. So you were either going to make one huge mess or you were going to specify it to the particular installation.

Pantages: What about standard reporting?

Hopper: It didn't matter what they did to begin with as long as the reporting set was coming. You can pull that off later anyway. That's a separate program.

You've got these engineers getting these small computers, one step up from programmable calculators. These aren't going to be used anywhere else. They're using it for what they do at the moment. If you put everything in, you'll have a colossal thing that no one can make use of.

Dick, I want you to hear the Raytheon guy, because if they've got something in storing modules and constructing programs out of it.

Burdette: Yeah, sure. I want to hear that. We're working on that too. The University of Kansas is involved in it.

Pantages: There's a lot of working going on in modular package development, isn't there?

Hopper: Oh, going back to the beginning of compilers. You know what the big failure was? Those of us who learned in the beginning to do modular programming and everything never passed it on. We were too busy getting programs on the computers. By the time we got around to it they had the big memories and they were splurging everything all over the place. Then we had to back off and invent modules.

Burdette: Are you talking about subroutines?

Hopper: Macros, whatever you call them. Modules.

Burdette: They can be done with models too.

Hopper: I think they've really got something. [Raytheon] will be here the 17th.

Burdette: I'm going up to U. Penn, Moore School, to look at Model 2, which we've got them working on for us. Model 2 is in this area, non-procedural language, which sort of models…We've looked into FOCUS.

Our position is that it's fine as long as you do it in a controlled environment. So we can see what it's like and then can evaluate it properly before they try to decide to do an entire system that is going to be at 85 sites. They want to go ahead and do it right now without proper evaluation.

[*The Burdette conversation ends here.*]

The Murray Siblings: Brilliant Communicators

Hopper on phone: I can't. I start traveling again. I begin the fifth of January. I'll be gone from then on. Merry Christmas and a Happy New Year.

Pantages: I know I have to let you go. We got caught somewhere between Eckert and Mauchly and the future.

Hopper: Any day is a bad day. This was a better day. Next Monday. Tuesday I have to be at CODASYL.

Pantages: To give you an idea of what I know about your role in this industry: inventor, implementer, leader of standardization, educator and communicator, visionary. There is no beginning and end with you.

Hopper: If you talk with my brother you'd find the same thing. When he graduated with me, he went to Banker's Trust. He started to get his doctorate, completed it just before WW II broke out. He entered the Army under the draft because he couldn't get past the physical to become an officer because he has eyes that have a blank spot. He was ordered up to the airfield up there. …And the sergeant that was writing the colonel's reports, he helped the sergeant. The colonel found out he had a doctorate in economy. So he was sent off to OCS and he was commissioned and eventually worked for Hap Arnold. He wrote the whole plan for the military and transport service for what they would do when the war was over. He went to [a] bank where he became a vice president. He found that he was more interested in teaching. He retired early, but the bank set up a Sloan School professorship at the graduate school of business and made him the Sloan professor of finance. He also started lecturing all over the place.

Pantages: What was his name?

Hopper: Roger Fletcher Murray. He lectured all over the world. He was an expert on pensions. He retired at 65 but he's still all over making speeches. I think we are one of the only pairs of sister and brother, both of whom received the Legion of Merit in the military, and received honorary doctorates and have doctorate degrees…and everything else.

And my sister Mary, during the war, she had small children so of course, she ran the nursery school at a plant at Bloomfield for all the women that came to work during the war. She had a double major too, and after the boys were off to high school, off she went to become a statistician for an insurance company. There were three of us, not one. All of us had the ability to get up and talk. It must have been in our background.

[Late in December, we resumed the interview by phone.]

Common Sense and Distributed Computing

Pantages: I have been talking to people about you and you were incredible in a lot of ways, but there's one way in particular—and that is that you have not changed, Grace. Your character, your attitude, your personality…

Hopper: Except that I am always after something new.

Pantages: [Laughter] Well, that's one of the characteristics that has never changed.

Hopper: One thing is that people try to make me something extra. I think they've failed to realize that everything I've done, the so-called big accomplishments – like writing the first compiler and starting the business about distributed computing and all of that. They totally fail to realize that everything I've ever done was not genius effect. It was all straightforward common sense at the time: how can you get this done?

Pantages: But it was also the fact that you were capable of recognizing it.

Hopper: I think I always looked for the easiest and best, most accurate way of getting something done. I don't use high-level theory. I use very basic common sense – the whole original concept of the compiler is the most common sense thing I ever heard of. And I've always said that if I hadn't done it, somebody else was bound to sooner or later. It was so common sense that it had to be done.

The same thing with the question of using parallel computers. We're getting more and more information and everybody wants it faster and faster, and you're obviously not going to do it with one computer. You have to build a system of computers.

I think that's the failure, the inability to recognize the need for constant use of common sense. Most people don't use it; they try to use some kind of a theory. I think they don't look at what the problem is and what's the easiest way of fixing it. And then you have the second part of the job, which is to persuade someone else that they need it and to spend the money to get it.

As I watch people try to get things, they try to sell everything. That's not the way to do it; what you do is figure out why the other guy needs it and then you tell him that. You leave yourself out; you explain to that guy why he needs it, why it is good for the company, if he sponsors this he might get to be manager or vice president.

Pantages: That sounds simple.

Hopper: Well, people don't look at other people and what are his interests and what he cares about and talk to that rather than what they think. They are so wound up in themselves and what magnificent idea they've just gotten. So when you come down to it, the answer is I've always been extremely practical.

Pantages: You started talking about small systems. Jack Jones, I talked to him yesterday, and he said that in his thesis on automatic coding in 1953 or 1954, he quoted you on the desirability of small systems and distributed processing, which is the earliest recollection of the discussion on that issue.

[*Jack Jones, the head of computing at Southern Railway and subsequently information systems vice president of Norfolk Southern Railway, was also a pioneer in CODASYL's creation and COBOL's development and remained a leader throughout CODASYL's existence.*]

Hopper: And the reality of the thing was the way IBM pushed the industry, partly for economics. I was on the other side of the fence.

Pantages: Yours may have been more realistic. But I think IBM had put dollars and cents behind the thing [*mainframes],* having figured out how to make the most money.

Hopper: But in the long run I've got the information to be done my way. And COBOL won, not PL/I.

Back to 1943-1949: Howard Aiken

Programming: Aiken's Idea

Pantages: We talked a little about your time with Aiken and Mauchly.

Hopper: Aiken was very much practical, the same as I was; he was an engineer. He didn't start out as a mathematician. He was an electrical engineer. Electrical power engineer as a matter of fact – before he had to write a thesis and found he needed a computer. He had gone back to Harvard to get his doctorate degree. He got interested in the ionosphere and bouncing radio signals off the ionosphere. He got into some nice complicated equations and he was pounding them out on calculators. And he wanted to study the ionosphere some more. He got fascinated by it. And so he ended up in computers. He had to have a computer; he couldn't do it all in his lifetime.

Pantages: Was that in the early 40s? And you showed up there.

Hopper: I was ordered in there. I think Aiken's first paper proposing a computer was in the 1930s, 1937 to 1939, and it's in that Brian Randell book. He was going to have a switchboard and all these different pieces hitched to it so he could perform the operations. It was almost a blueprint, a background, for Mark I.

["*The Origins of Digital Computers: Selected Papers" edited by Brian Randell (Berlin and New York: Springer-Verlag "Text and Monographs in Computer Science," 1982)*]

And I think we've neglected him, because his computer was programmed. It may have been built out of relays and step counters, but it was just programmed the way we do it today, step by step. ENIAC wasn't, ENIAC was plugged together. And I get awful tired when everyone credits Von Neumann with the programming idea because it was Aiken's, and Von Neumann had been up there to run problems. And when it comes down to the actual question of construction code for computers, basically in the heart of every computer is John Mauchly's construction code. It was Mauchly who designed the C1 code for UNIVAC. That UNIVAC I code is the heart of every code that's ever been written for a computer, they just put bells and whistles on it.

It had "bring," "add," "store," "multiply," "divide," at the heart of every machine code. That's the single address code.

Pantages: Was Aiken conscious of (his impact on) programming concepts?

Hopper: Yes, you know he built a coding machine for Mark III, and probably the reason we never hear about that is because it was mathematically oriented, not data processing oriented. And once you sat down at the coding machine, when you wanted to put a sine program in your program, you pushed the sine button. The coding machine was a relay machine that wrote a program. You had lots of keys and you just pushed the symbol you wanted. So he was definitely concerned with making it easy. You might say that's where I got part – at least – of the concept of a compiler.

What the coding machine did is check its relays and enter the subroutine in the program. Only I wanted it to be more flexible and not have the all-purpose ever-complete subroutine in my program. I wanted to modify it on the basis of the information I had.

So many of the concepts that we live with day in and day out today were latent or evident in those early days.

Pantages: Were there others?

Hopper: For instance, Aiken did run the Prudential problem, using two digits for each letter and showing that you could run data processing on these machines.

Pantages: Tell me about that. Did they come to Aiken with the problem or was it just fiddling around to see what would happen?

Hopper: I don't know which one initiated it, Aiken or Prudential.

Pantages: Were they looking for a way to use the computer?

Hopper: They were looking to see if the computer was going to help them because they were running out of punch card machines. Of course, they were among the most advanced in the use of punched card machines. And here was a chance of sequencing without picking up the cards and walking across the room with them to the next machine. It would be continuous. So I don't know if he went to them or they came to him. But up in the Aiken library there's a copy of that original report, because I left my copy there.

Pantages: Is that what sparked your thinking in terms of languages that would make it easier to use?

Hopper: That was after I got down to Eckert and Mauchly. I didn't really get into data processing until I got down there. And of course they had designed to build UNIVAC. It was built for data processing.

Pantages: Let's go back to Aiken and Harvard and some of the people you worked with. You worked with Bloch.

Hopper: Bloch and Bob Campbell were there when I got there. They were both ensigns.

Pantages: I didn't know Bob Campbell.

Hopper: He's now I think at MITRE up in Boston. Bloch is up in New England, Boston, he works as a consultant. Bloch leaned more toward the hardware side, and Bob more toward the programming side. Yes, I said Bloch is the only person I ever knew who wrote a complete program in ink and was correct the first time. But Bloch just thought like the machine did. He was a mathematician of course, that was his degree.

Pantages: At that point, everyone seemed to come out of mathematics and engineering into computing because they had problems to solve.

Hopper: Most of the problems during the war dealt with ordinance of various kinds. It was during the war that we got the proximity fuse. We got the rockets, the things that were self-propelled, instead of just giving them a shove and letting them go. Actually the Navy had a couple of guided missiles during the war, and the whole world of attack was changing completely.

Pantages: The problems you were running, then, were in the design area?

Hopper: That and for instance, they developed magnetic mines, where the big lump of steel would detonate the mine. Then you had to be able to sweep the mines. One of the problems I ran was towing a die pole behind a ship. How far away would it detect the steel?

How big was its environment? Nobody knew at the time. So they were connected with new weapon systems basically.

Pantages: Were you working with other developers at that point?

Hopper: That's how von Neumann got up there. There were lots of men in and out and our course for the radar school was just across the campus from us – there were problems from that. For many of the problems we just didn't know what their application was. We were just told to make tables of certain functions. Practically everything was connected with the war. It wasn't until after the war that we tried anything else. We had a hotline straight from the laboratory down to the Navy Bureau of Ordinance.

Pantages: You left there in 1946.

Hopper: No, 1949. I went to inactive duty in the Navy in 1946 and stayed under a three-year contract with Harvard, which was under the Navy contract. Navy had the contract with Harvard to build Mark III, which was later taken over by the Air Force.

Aiken: The Value of Writing and Documentation

Pantages: Incidentally, you said Aiken taught you the value of documentation, while teaching you how to write. People since then have said he put a very high value – logical thing to do – on documenting everything he did.

Hopper: He sure did. And what's more he taught me the best way to get something written, which I'd never thought of before – he told me to read it aloud. He pointed out that if you stumble when you try to read it aloud, you'd better fix that sentence. Every day I had to read five pages of what I had written. And if I got stuck I'd go back and do it over again.

That stuck with me ever since, so that instinctively I read it almost as I write and certainly read it over before it goes anywhere so that I know if I stumble I have a bad construction or the wrong word or something. And I immediately start to fix it. I think that's one of the most important things about learning to write that ever happened. I've taught it to all the kids that have to write reports for me. Read it aloud.

Pantages: I wish I had taken a writing course from you. It took me years to learn that.

Hopper: It does make a difference. It also keeps your sentences clear and simple, because you have to breathe at the right places. You don't write this long turgid mess that most theoretical people write. He also taught me that if I'm going to give a talk I should tape it and listen to it. If it gets bad at some spot, I might go back and do it over again. I taught my kids, I kept a tape recorder in the office, so if they are going to give a presentation they could tape it and listen to themselves and make changes.

He taught us all how to communicate. I've come to feel that there is no use doing anything unless you can communicate. And I include that in my talks. One of the most important things we have to do with our young people is teach them to communicate.

Pantages: Probably a lost art at this moment.

Hopper: It's amazing what you can do. Now George came in there from Texas and I went to work on him naturally as I do with anyone that works for me. I've also felt that when people work for you and do something, you give them the credit, not you. So, when we had to give presentations to higher echelons, the committees and stuff, I'd take the whole crew along with me and then I'd present each one to give his section of the stuff. So they had to learn to speak. I'm not sure that George would be a GS-15 today; he came in as a third class petty officer.

George is George Baird. He's the one that's now head of the Federal Compiler Testing Service. He'd just lap that stuff up. He came with a group in 1967 as a third-class petty officer. And he wrote all those beginning routines for testing COBOL. And the one thing he needed was to be able to present it. I gave him the practice and now he is a GS –15.

Pantages: Howard Bromberg told me, "The first talk she made me give was a three-hour lecture at St. Paul."

[*Howard Bromberg, who worked with Grace Hopper on program development at Univac, was on the original CODASYL Committee from 1959 to 1962, and hence one of the authors of the original COBOL. Moving to RCA in this period, he worked on the compiler for the RCA 501 for COBOL at the same time that the UNIVAC compiler was in process. He became the first chairman of the ANSI Standards Committee for COBOL*]

Hopper: I've always felt that the youngsters who work for me had to get the credit for the work they had done. And that they should be given the opportunity to present it. Also if they presented it, they'd get the questions and I couldn't answer the questions anyway. And I always felt you had to give the credit to the people who really did it. And I've seen too many people – professors and businessmen everywhere – where the boss, even though all he'd done was to tell other people what to do, took all the credit for it. I never thought that was fair when you had people working for you who were developing things and inventing things.

Anyway, what I did learn was the more you give it away, the more it always comes back to you in the long run anyway.

***Back to 1949-1964:* John Mauchly and the Univac Years**

Pantages: Let's go on to your years with Eckert and Mauchly. You chose to go there because of John Mauchly.

Hopper: Well, he was the one that hired me. The other reason was that there were two companies that were going to have working machines that year. One was Engineering Research Associates (ERA) in St. Paul and the other was Eckert and Mauchly in Philadelphia *(Eckert-Mauchly Computer Corporation*). And I liked the idea of Philadelphia better than St. Paul. And out in St. Paul was my good friend Howard Engstrom, who was one of my professors at Yale. He was president of ERA then.

So there was a strong tug either way. Of course it ended up eventually that Remington Rand – Jim Rand – bought ERA.

Pantages: Was Engstrom still there when they bought it?

Hopper: No, he had died by then, or just after that. See, ERA was the old Naval Communications Annex, which built the relay computers that did the coding and decoding during the war. And that's why the Navy helped them find some money to support it and start a company. And Engstrom had been the captain of the Naval Communications Annex, and William Norris was the commander and executive officer. And of all of those young officers, eventually Robert McDonald* became president of Univac.

[*A few years after the Sperry/Remington Rand merger (1955), Jay Frank Forrester, a top Sperry executive, succeeded William Norris as the president of the Univac division. Robert McDonald succeeded Forrester when he became president of Sperry Rand in the late 1960s.*]

Pantages: Was McDonald out of that group too?

Hopper: Yes. They are all still at Univac.

Pantages: An illustrious group.

Univac – Navy Men and "Bright Young Women"

Hopper: But having everything classified, most people never knew them. And of course when Sperry took over, Frank Forrester was a Naval academy graduate. So there's always been a tinge of Navy all through Univac. What happened was you had the original Navy group, and then, if you had two people in front of you with equal ability and one's Navy and one's not, you'd hire the Navy man.

Hopper: See Mauchly had been building his team for ENIAC during the war. The bright young men were all in uniform and were in France or the Pacific. And he started hiring bright young women. I'd say close to half of the people that worked on all the beginning work on programming and the non-hardware side of ENIAC were women. And his right hand assistant in building the C-10 code was Betty Holberton. And he encouraged her to write that first sort/merge generator, which was the first time anybody used the computer to write a program. And part of it was necessity. The young men weren't available. And the bright young women were.

But he never changed, even after we were bought by Remington and Sperry. The attitude at Univac has always been different from the other companies. They've always been there.

He started what's still a Univac custom – every spring there was a course in programming which is open to anyone in the company. That's how so many secretaries got to be programmers before we were through.

Pantages: That must have been unique at that time. Someone else said that to me, Bill Crowell, that you pushed the hiring of business people for programming for business applications.

Hopper: They were the guys who knew the problem. You didn't want engineers trying to do banking and insurance. They didn't have the vocabulary even.

Pantages: That's only common sense. They are still talking about, "We've got to start hiring people who know the field."

Hopper: You're right back again to common sense. A gal who was a good secretary was bound to become a programmer, meticulous, careful about getting things right. Step-by-step attitude. The things that made them good secretaries were the very things that made them good programmers.

Plus I think that it's always been true that women were more willing to finish the job. They'll stay with it; tie up all the loose ends. If they make a dress they put the snappers and button holes in. Whereas men, once they've solved the problem to their satisfaction, want to go on to the next problem instead of tying up all the loose ends. So you usually get better documentation from women. You can't make that the final generalization, but I think it's pretty much true. They are more willing to complete it and tie it up in a neat little bow at the end.

Then I always said that the concept of getting the data all together so you could operate on it was the same thing as getting a dinner ready. You had to get all the parts together and have it finished at one time.

Another thing… I was lucky that each year I did my two weeks training duty. And usually that meant I came down to Washington and spent two weeks learning the latest things Navy was doing. And that was an assist all the way along the line…constantly keeping me up to speed on what was happening in the Navy.

From those 20 years I was at Univac, I saw the development in industry through Univac, and the development in government through the Navy. I didn't get isolated. And of course I was always teaching. So that left me in touch with the academic world.

Pantages: What course did you teach?

Hopper: It was a seminar on computers, new developments, future, usually. A little bit of history, a little bit of now, and a little bit of the future. There's always a little bit of management in the background, but never explicitly. I've also always hit the need to continue to offer training to our young people. I think the sad thing right now is that when people start cutting back, the first thing they cut is training and books and periodicals. And in an industry that's moving as rapidly as this one is, it's terribly important to keep people up to speed. Otherwise, they begin to die on the vine.

Now in this transition period, they also mention that one of the places to cut is on travel. But in the travel budget is the money that sends people to seminars and the meetings. And that means we are cutting our people off from learning what's going on in their field. They always think they are going to stop the high mucky mucks from traveling around the countryside, but what they do is cut off our ability to send people to conferences and seminars and meetings.

Pantages: Profitability is in new ideas and it always seems strange that they run counter to that.

Hopper: I think that the people who cut the travel in the government hear so much about the high mucky mucks who commandeer a plane to fly around the country for some reason or other. They don't hear about the junior people that are being sent to courses, for instance, about the whole business about database machines. The first seminars were given at places like Utah and Kansas State. You need travel money to get there.

They don't see the low-level travel and what it's for. And when you put a new computer in, you've got to send someone from headquarters to look over and see that they got all the right things. And the inspection trips are cut back. One of the things I always did if I got put on one of the inspection trips was see that they set up a library, which is awfully important. At least they are getting free periodicals.

Battling General Purpose Software

Pantages: Let’s go back to the Eckert and Mauchly era. I am aware of the compiler developments there, but there were some arguments – interpretive techniques vs. compiler techniques. You mentioned that you and Mauchly didn’t see eye to eye on some…

Hopper: Well it wasn’t so much Mauchly as it was Tolly [Anatol] Holt and that group. They built an entirely different kind of compiler in which you could constantly generate new code which was fed into the compiler. My point was that it would become a monstrosity. Every time they wrote something new using some of the old subroutines or something, it was made a part of the system. General purpose coding, GP they called it, I think. And it was just going to mushroom forever. And you couldn’t do that. We’d get beyond the capacity of the machines, sooner or later. It was a nice idea, but… I also objected to generating code at object time. Because then every time I ran the program, I’d generate the code again. I wanted to store it and use it over again.

Again, good old Scots attitude, I didn’t want anything in my final program that didn’t pay off for me. That’s why I went after the operating systems. Because it was perfectly clear that they contained general purpose for everything. And then in running any one given program, that was not what I needed. That’s why today I’d like to rip our compilers apart and have a generator for handling the data and a generator for this and a generator for that and get it all done ahead of time until I had the specific program for the specific job. Nothing general purpose left in it. You could build a generator that would write you a particular code for the particular types of interrupts that you are going to get in a particular problem… Instead of using a general-purpose interrupt handler.

And at object time I would like to execute only what was necessary for that problem. I am still fighting that one – because we are going to have to go faster and we can’t afford to use general-purpose stuff when it’s possible to write special purpose.

Pantages: Again, are the economics viable for that?

Hopper: Yes. Sure we had things like that built into our original compilers. The original UNIVAC II COBOL, for instance, generated the specific subroutines for handling the input and output data for that specific problem when you compiled the program. It had an I/O generator in the compiler. You can afford that because you only compile once. That’s only one step in the thinking. You are going to use the program again and again afterwards. So it has to be the tightest thing you write.

Pantages: Yet IBM, for instance, has been pushing “hardware prices are coming down, don’t fool around with the software. “

Hopper: They want to sell you more hardware. They aren't going to make any money in the long run… [as much] as from renting hardware. The more the program sprawls the better off IBM is.

Pantages: Except they are starting to charge for the software now, but they are still getting away with those practices.

Hopper: They don't charge for the run time on the software. My aim was to cut run time. I really think that the future holds more and more information that they need for faster and faster answers. The whole future looks that way, and that's the direction you have to aim in.

Pantages: When you talk about speed, I conceive of the need for speed for weather forecasting and other massive applications.

Hopper: Suppose you have a salesman in Kansas City and he's got a customer who calls you up and wants to know if you've got five large anchors on hand. You need speed. You've got to be able to get into inventory and find out if you have five large anchors and tell him so. You can't afford to have a clunk, clunk, clunk program that's going to keep him sitting quiet on the end of the phone for 10 minutes. Even inventory will have to go at a high rate of speed.

Pantages: Don't we have technology fast enough now to solve the vast majority of problems like that?

Hopper: Yes, we're beginning to get there. One of the things that Westinghouse tackled right off the bat was a close-to-online inventory. But they are slow compared to what we need today, and that of course is the reason I wanted to pull the inventory program off the mainframe and put it on its own computer. And I finally realized that the only reason we put inventory and payroll on the same computer is because we only owned one computer. A logical common sense reason why they should be there. So let's pull them apart and they go faster if you get them on their own computers.

Plus the fact that the dollars are pushing in that direction – that a system of small computers could cost a lot less than a big computer and you could be rid of a lot of the overhead. Because each one could be dedicated to its own job, it didn't have to be general purpose.

So see all of these things are ending in the same direction: pointing toward individual computers for the individual job.

Pantages: In the context of time – back then...

Hopper: Back then the cost wasn't one of the influencing factors. It wasn't until we began to get the chips and the small computers that it became economical as well. But there was a question even then how much we were willing to pay for an instant answer. And some companies were beginning to become ready to – because it meant sales and all sorts of things.

I think that too often we look at the computer separate from the rest of the company, and you have to look at it as integrated, as part of the company. And if the speed of the reaction of the computer helps sales, then you are going to go for it. If you only look at the computer all by itself, you may not pay any attention to that. But you have to look at it as supporting sales and shipments and everything under the sun. You can't just leave it off in the machine room. It interacts – not just for processing data, but it interacts for speed and response and everything like that for the rest of the company, and the kind of the records you keep and everything else.

Integrating Computers into "The Whole Company"

Pantages: Where do you think we are now?

Hopper: I've used the analogy of the Model T, which is what made a tremendous difference in the development of automobiles and transportation in this country.

I think we have the Model T computers. By analogy we are at the beginning of what will be the world of computers. We've been sort of through a bunch of preliminaries. Now we are really ready to begin to use computers.

Pantages: You are talking about end use and true integration through an organization

Hopper: Where they become an integral part of the whole company.

Pantages: You are saying that even though they have been talking about this, they haven't really been doing it.

Hopper: No. Well for one thing, there still aren't enough people who can do that systems design, who can look at the total flow of information through a company and then design a method of implementing it. We began to get those real systems design methods when NASA started work. And their whole concept was that you took anything and broke it down into subsystems and defined the subsystems and the interfaces between them. It's just what we need to do with the systems in the company. And it's non-linear of course; things do happen in parallel in any given company. Sales is working the inventory at the same time Personnel is working.

But it means changing people’s minds. It means upheavals in some organizations. The big speed-up will come when the next generation of young people come who have grown up with this stuff. We still have too many people at the tops of things in awe of their great big computer. And they’ve been so busy that they haven’t had time to find out about the new stuff.

Pantages: Changing people’s minds runs straight through your career. Everybody that I talked to said you were very good at that. But you ran into some stumbling blocks.

Hopper: You always run into some stumbling blocks. There are always people out there who are screaming about the future, and they just live out there and yell about the future and try to explain to people why it is to their advantage to do things differently. And it takes you a little time to change people’s minds. You have to show them why it’s going to be to their advantage.

COBOL’s Ancestors and the Rise of Data Processing Programming

Pantages: Going back to your time with Eckert and Mauchly and then Remington Rand – Remington Rand wasn’t stodgy, but it was a company that wasn’t conscious of programming.

Hopper: Well, Art Draper was and he was sent down from Remington Rand as the manager in Philadelphia…of the Univac operation. And he listened.

Pantages: What happened? At this point, you came in and you were working for Mauchly. And then a couple of years later, Remington Rand came in. What happened during this?

Hopper: They split the two groups, and Mauchly’s group went toward the mathematic engineering side and my group went toward the data processing side.

Pantages: When you went to work for Mauchly…

Hopper: We were still doing mathematical scientific engineering problems. But it was the influence of Betty Holberton’s sort/merge generator and the beginning of the swing toward the data processing problem. The first one I did was on finding out how long extended insurance would go on once someone stopped paying their premiums. In life insurance they still do that. In most policies, they use the cash value to pay for extended insurance until that runs out. Then there was a slow but sure swing toward data processing. And I found the data processing more interesting. I had all the math and everything, but the data processing was more fun because it had people in it, and because you had things like union contracts that made sense but were not logical from a mathematical point of view.

Pantages: What customers did you work with?

Hopper: One of the first ones we went to was Carborundum, putting in the computer.

Those were all started with UNIVAC I.

Pantages: Earlier on, what was it like working with Mauchly? What kind of influence did he have? What arguments did you have with him?

Hopper: I don't know as we ever really had any arguments about it.

Pantages: The reason I said that was that you said you and Mauchly didn't see eye to eye about compiler development.

Hopper: His group was heading in one direction and mine was heading in another. There was competition there. But it wasn't from Mauchly; it was the people working for him. People like Tolly Holt.

Pantages: What was Mauchly like in terms of the things you learned from him, exchanged with him?

Hopper: Oh Mauchly was always willing to try anything. If it sounds at all reasonable, try it. So many people say "no" to begin with. He never did that. He always encouraged you to try things. If you had a bright idea, try it. He let people try things, let them have computer time, and time to do things. He encouraged innovation, the technical terminology I guess. He knew we were in a new world. He had full recognition of that. And he never let personalities bother him. He was nice to everybody, encouraged everybody, he was a good leader.

Pantages: Were there concepts you picked up from him?

Hopper: I don't think of any.

Pantages: Your group was the Automatic Programming Group.

Hopper: I didn't name it that. The sales department did. I didn't feel it was automatic programming because the individual still had to design it. It just made the computer do it, that's all. I always disliked that phrase. We still had to tell the compiler what to compile. It wasn't automatic. The way the sales department saw it was we had the computer writing the programs; they thought it was automatic.

Pantages: It was a glamorous concept.

Hopper: Typical of sales.

Pantages: Now, you were a saleslady.

Hopper: Only of ideas, though. I've always seen computers as a tool, as much as a screwdriver or a lever or anything else

Pantages: Someone told me that the way you got management to accept an idea was to go out and get some users interested in it and they'd sell it for you.

Hopper: I did that one once. US Steel, Westinghouse, Metropolitan Life thoroughly backed the idea of English-language programs. We might not have done it if we hadn't had them with us.

Pantages: What else was marketing management reticent to do, because they couldn't comprehend it?

Hopper: I don't think that it was so much they couldn't comprehend it… because they had been selling punch card equipment and typewriters for years. I think nobody knew what it was going to be like to sell a computer. I don't think anybody had any concept before we really got started just how much it was going to take to write the software for any given job, and I don't think anyone fully realized how much the maintenance job was going to be. I don't think the whole impact of the computer was fully realized until after they got out there. And I don't think anybody realized – either in the companies, not just Univac, but IBM and all the rest of them – what an impact computers would have on management of the company that started to use one. And thought that through. So there were an awful lot of things to cope with. Because it not only affected the people who ran the computer and wrote the programs, but it could ultimately affect the structure of the company using the computer. I don't think anyone was aware of that ahead of time until they got out there. It had to happen before you learned to cope with them.

Pantages: How did your thinking evolve in that time? The English-language compiler was a natural demand from your standpoint.

Hopper: I think I always recognized as soon as I started working with people and talking with people that there were some people who were totally oriented and perfectly happy using symbols, and there were other people who were not. That's where English came from. You talk to people in general and some of them will take abbreviations and symbols and manipulate them like mad and others won't at all. Some people get to be mathematicians and some get to be managers, or something. There are two kinds of people, some symbol-oriented and some not. Sounds simplistic but it's really true.

Pantages: It's true – some will talk in acronyms and some will tell you what they are talking about.

Hopper: That's right. So there had to be two kinds of compilers. One was for the symbol-oriented people, which turned out to be the mathematicians and engineers and all those people. The others turned out to be "the other kind of people" and you had to give them both the ability to use computers. I always wanted to let people use computers, which I think is such a nice step forward.

Pantages: Somebody mentioned that there was a point at which you hit a wall, which is one of the reasons you moved on to the research group at the University of Pennsylvania. Is that true, you had a disagreement and moved out of line management?

Hopper: No, it was more the fact that I recognized and they recognized that I did not want to go up in line management. It was management of research and development that I wanted. I did not want to go up in company line management. I didn't feel I was qualified for it, and it wasn't what I wanted to do. I wanted keep on playing with machines.

Pantages: Okay, so it was time for the programming group to start becoming more complex…

Hopper: And it was time to pull out the research and development of the programming group. And concentrate it. Pull that away from the line development of the software that went with the computer.

COBOL and CODASYL

Pantages: What were you working on at that point that you felt needed that kind of concentration, the three or four earmarked developments?

Hopper: The real development of what COBOL is today. The very first COBOL was pretty naïve when you think about it. And the first COBOL, the early COBOL's, were built for the dedicated computer, where there was no such thing as interrelationship with an operating system or anything else. Each program was a unit in itself and it ran all by itself. And we began to come up to the point of the interactions, and the direct access and all those things that were beginning to appear. And the compilers were going to have to change.

Pantages: Where are we now, at post COBOL 60?

Hopper: This is where we step forward into the real COBOL we have today: where you have things interacting, things happening simultaneously. And we began to think about systems and computers and communications between them. This is where communications really comes in, where they aren't just on an access, asking a question, but beginning to really interact.

Pantages: Is this what you took your group off to do?

Hopper: Yes, we were beginning to look toward the future. People were beginning to have three UNIVAC IIs instead of just one. The whole thing was no longer monolithic.

Pantages: Univac was of course one of the real pioneers…

Hopper: Has been all the way along the line, which people have forgotten

Pantages: …but in real-time and on-line programming.

Hopper: And that influence was coming in from the Navy on account of the nuclear submarines… where you had three computers operating simultaneously and they had to compare. That influence is beginning to penetrate, and of course, there again I was getting the feed-in on my training duty, and that reinforced the idea of multiple computers.

Pantages: What was your role in this?

Hopper: As far as the Navy was concerned I was a listener. What would it do in business. I was listening to everything under the sun, from every direction at all times and reading everything I could get my hands on. And then trying to see how I could put them together and use them in the business environment. Being alone I spent a lot more time reading than most people did. So these things were coming at me from all directions and I was trying to figure out how to use them.

Pantages: Howard Bromberg said he complained to you about that. He said you would ride home together in the evening and you would talk about a problem that needed solution. And you'd discuss it a bit. He said, "I'd go home and read a newspaper. And I'd pick Hopper up in the morning and at 8 o'clock she would have the problem solved. I told her she was taking our problem-solving fun away from us."

Hopper: Again it was a constant reading in all areas. People sort of stayed in separate boxes, but if you combined the boxes, you had the answer. I didn't invent the answer. It was being aware of everything that was going on and combining the pieces.

Pantages: In parallel – at the same time this was going on, CODASYL – it started a few years earlier, right? And you were involved in that.

Hopper: Yes, it started in 1958. But the leadership of that was Cunningham. And Jack Jones and Al Ash, the Air Force.

Pantages: That goes back to the development of COBOL itself, right?

Hopper: That's right.

Pantages: Your role was to implement the specs they came up with.

Hopper: We'd written FLOW-MATIC before that, and if you take the FLOW-MATIC manual and compare it with COBOL 60 you'll find COBOL 60 is 95% FLOW-MATIC. So the influence of Commercial Translator in fact was extremely small. But I figured the thing to do was corral those people and when we had something to say, we'd say it was a compound of FLOW-MATIC and Commercial Translator and keep the other people happy and wouldn't try to knock us out. We'd give them some credit and they'd have to get on board with us. But if you compare the two manuals you'd find that it had hardly any influence at all. But if you gave them credit for it, why they'd go right along with you. If you didn't, they'd fight you. You can always give credit, you can always afford to.

That again is the practical. Think about the other guy and his position and his interest. You are always trying to work with people rather than against them. You've got a new idea; give the boss credit for it. It doesn't cost you anything.

Pantages: COMTRAN was an IBM development wasn't it? That's what they wanted rather than COBOL. So that's the way you got around them.

Hopper: Yeah, give them credit for it. Meld it into COBOL. Bob Bemer went along with it. He was the original representative to CODASYL for IBM. He'd gone back as head of the scientific engineering FORTRAN group and Bob Bemer was the one doing the data processing side and he and I were originally technical advisors to the CODASYL committee.

Pantages: Are you still an advisor?

Hopper: No, they decided that was giving too much favoritism to Univac and IBM. Eventually I was actually a member… after I stopped working for Univac and became a user. I'm on the executive committee. I was the Univac representative on the COBOL committee, until I found that I was fighting for my ideas instead of looking at the whole picture and decided they better put someone else on. You can get too wrapped up in your own ideas.

Pantages: People have said those two things about you – not that you get wrapped up in your own ideas, but you are very strong for your own ideas. "The problem with Hopper was that she's right most of the time." On the other hand – and people would turn around – I think you very successfully carried this off – they would say, "She is always willing to listen to someone else's ideas." So you are very successful in defense of your own ideas, but people still have the sense that you will give them their day if they can prove it to you.

Hopper: Yes, because some times I get out of date. I've got to listen to the new ones. When someone comes up with good ideas, I will sell theirs as hard as I sell mine. When we came to the sub routines, it was George [Baird's] ideas of how we could fix our programs to run on anyone's computer. And I've been selling that every since. Because most people haven't realized it yet. It was George who wrote the program – when we wrote the COBOL program, he invented the way by which to used Xs, x sub something for the special names and control cards. And then he stored little files that contained those. So he just told the routine which computer was one and he stuck the special names and control cards and off he went. That was totally George Baird's convention. And people still haven't fully appreciated that. Those test routines for both COBOL and FORTRAN would run on anybody's computer and they have – by sticking to the standard language in his technique for handling the control cards. George is the one that came up with that.

Pantages: That set a pattern for the industry now, at least for the government.

Hopper: So now George is a GSV 15 and manages the compiling testing service, which he fairly earned. And that fitted beautifully into the whole concept of portable programs, which is one reason for having standard languages anyway. There were two reasons; one for the training of people – you could train them once – and the other was the portable programs.

The Push for Standard Languages and Education

Pantages: Put me back in context historically. And note that you went back to the Navy in 67. When did your interest in the support for a standard begin, in other words the testing and validation?

Hopper: That was then. Norm Ream felt we had to standardize the programs and be able to move them from one system to another. And the way to do it was standard languages. And he also felt that the only way we were going to get standards was – he was thinking of course of the Underwriters Lab which tests things and say they meet the standard, and there had to be some tests to meet the standard. That was when he asked me to come back and do it. And I said I wanted programmers to help me do it. I got Ed Ford, a civilian, and we had one young lieutenant and two sailors. George Baird was one of the original sailors. And Arnold Johnson [*National Institute of Standards and Technology],* who is out there as deputy director, was another one – he came in the second group. They were DP3s? That came in. And all the development since then has been done until they broke the compiler testing service off and put it off by itself. They all came in as young sailors in the Navy.

Pantages: Is the testing service now under DOD or GSA?

Hopper: GSA. And that was done when all of a sudden the Appropriations Committee discovered that the Navy was performing a federal function. So they picked it out of the Navy and put it over in GSA.

Pantages: Navy has always been avant garde. I remember the PCM effort out of the Navy.

Hopper: There's always been an environment in the Navy that is a little different from other environments. You were on a ship at sea with no radio communications, and you've got to be totally self-dependent. That forces a lot of developments that you wouldn't have to make if you were going to land at an air base or return to headquarters. And of course the atomic submarines pushed a lot of new ideas. The need for the tremendous computer abilities there. I would say I think for new concepts we've gotten more from NASA and the Navy than almost anybody. Though they sometimes lagged in the hardware they put it on.

But NASA has had a tremendous influence recently. After all, the chips were for NASA. That's what made it possible to really begin to think of systems and computers. It's all been thought of as a flow, with other streams feeding into the main flow as time went on.

Pantages: When you began your research and development group, you also went heavily into education.

Hopper: There's another realization – not just mine but also Univac's – that there was going to be a need to support the training of computer people. And Univac always made major efforts – I'm not the only lecturer they sent around. There's Carl Hammer,* and Harold Joseph who would go around and talk to all the societies. Because one of the future great needs was people not only for Univac, but also for all the customers. And, when people began turning in their UNIVAC 80s and 90s way back then, an awful lot of those were reconditioned and given to schools and colleges. And I think there's always been a realization at Univac that part of getting ready for the future was to train people and to support schools and colleges.

[*Carl Hammer, with Univac for 20 years, primarily as director of computer sciences, was a highly regarded educator and leader in the development of the industry's professional organizations.*]

They've always done a great deal in that area. Quietly. I don't think they've ever gotten the full credit for it. But there's always been a push and an interest at Univac in training people. Even the course given their own people was important. They realized there weren't going to be enough people.

From that point of view, you have to add that Univac has always been a much more people-oriented company. When McDonald was headquartered in Blue Bell [PA], anybody could go talk to him. You could always go talk to the boss. Very open. Very people conscious. And I don't know what company has done what Univac has. Twice since I've been retired, they've upped the pensions for everybody. And they don't have to because there is no contract or anything. And twice they've increased the pensions for people who have all retired. They don't have to do that and people who work at Univac hope when they get retired, maybe the company will be generous when they retire and they'll get an increase in pensions if inflation dictates it.

Pantages: Howard mentioned that you got him a difference in pay when he went into the service for which he is still incredibly grateful for.

Hopper: Well you see, again there, Univac had always been service conscious because of its close relationship to the Navy. It's a more human company. And somehow it's continued that way. That starts with the president.

Pantages: IBM is people-benefits oriented.

Hopper: Yes, but not people oriented. The amazing thing is the number of people – as I watched over the years – who've left Univac and have later come back again.

Pantages: The problem I always saw with them was their marketing decisions, rather than technology or people.

Hopper: The marketing decision was controlled during the early years by the old Remington Rand group. And that's where the mistakes were made. Not in the Univac group. And it wasn't until we got Forrester that the whole thing began to get really united. When Sperry came in the picture. They of course brought in more of the technical side and the marketing began to change. Up until then St. Paul and Philadelphia had happily gone along building machines and everything, but the marketing was managed out of New York, out of the old Remington Rand headquarters. It wasn't until Sperry came in that you began to get a more technical orientation into the market. And a more realistic point of view.

Pantages: It always seemed to observers that they had edges technically that they didn't capitalize on them the way they should have, such as in the communications area.

Hopper: Yes, that was right. They had to get some of the technical people into marketing, and get some of the marketing people to know more about what was being developed. It was Forrester and McDonald who brought that about.

There were a number of mistakes that could have been made. I think it was remarkable there were so few. Well, when you look at GE and RCA, I think Univac did pretty well.

Pantages: Especially since Univac is one of the companies that had other interests.

Hopper: Sperry still makes great hay balers. And hydraulic controllers come from Vickers.

1964-1967: University of Pennsylvania, Moore School of Engineering

Independent Modular Systems vs. Big General Purpose Computers

Pantages: Between 1962 or 1963 and 1967, that's the time you took the research group over to the Moore School at University of Pennsylvania.

Hopper: Yes, I was beginning to think about combining things – the interrelationships in the systems of computers. I don't think I spoke much about it until 1965-66. That's when I was beginning to think about getting rid of operating systems, and I was out in the edge of nowhere again.

Pantages: Explain that.

Hopper: The more I looked at an operating system, the more I could see that it was made up of … the whole 1108, which I looked at after it was designed, Exec 8 was modular. It was one of the first modular operating systems. And the minute I could see that it was modular, I could see that each module was general purpose. And when I ran my program I didn't need all of it in there for any given program. I only needed about one-third of the operating system. Then the fun began of how to get rid of it.

Pantages: You had to have been fought on that one.

Hopper: And they still are. Everybody's still loading in huge operating systems. One of the ways to solve it of course is to get inventory on its own computer and then you don't need an operating system. You go back to a system of UNIVAC I's that communicate with each other. I think that's one of the reasons it's hard to get this across. We don't have many people around who ran the early computers, who know what you can do with a standalone computer.

What I think in terms of today is that I know what UNIVAC I did with only 1000 words of storage. And therefore I can conceive of taking these micros and building a system, each one of them doing a major job.

Pantages: When I came in it was like, 16, 32, 64K memory. And I kept saying look how much they did on those systems. You were conceiving of the original systems as effectively minis.

Hopper: Just translated it. That 16-bit chip I've got is awful close – the new 68000 Motorola is very close to being as powerful as UNIVAC I. So all right, I can have a system of them. On the other hand, I think marketing, and the other people in design, are scared to try and…I think there's a tendency to sell a guy one box, so they are trying to put them all inside the box. I was going to let them be standalone.

When you look at the UNIVAC 80 and you open the door, there's an awful lot of computers inside there. Interpreting instructions and layers and all sorts of things. And by now on every disk and everything, you do have a computer in it. But they aren't telling people that yet. They think it'll scare them.

Pantages: Do you remember the original proposal Univac made to United, the one that failed? It was trying to do all things on one machine more or less. They had a half dozen of them or something. That and the Burroughs and TWA effort.

Hopper: That's right. The thing that's going to get us into that future is not theory. It's not going to have exponents and subscripts and high mathematical theory. It's going to be common sense. And that's awful hard to sell to people. It doesn't look pretty. When you write it out, it's just some words. No exponents. No subscripts. It doesn't impress everybody. So you've got a job there.

Pantages: It's tougher to service because you have to go so many places, rather than one.

Hopper: No, now that each company has all the pieces you don't run into that. Besides which the micros don't go down. Reliability is fantastic. That darn little Motorola of mine didn't go down six years. The only things that go down now are things that are electromechanical. You still have problems with disks, burners, but not with the computer anymore. Besides which the price of the chips is coming down so I wouldn't be at all surprised to see at least one company sell you a database management system and you buy the software and they throw the hardware in with it.

Pantages: It looks like Cullinane is doing that.

Hopper: That's it exactly. And Cincom too, probably. So we're going to turn around in the other direction again. The one thing that bothers me, for instance, here you've got calculators. You can have a programmable calculator with built-in routines for the sine, cosine, tangent and all those things. Why hasn't anyone put all those things into the big computers?

Pantages: They haven't?

Hopper: No, only Hewlett-Packard is thinking of it.

Pantages: I guess there's something wrong in my background. They haven't always had that there?

Hopper: No. You wrote the program for them. You have a calculator that has chips that have those things that have been worked out now for about five years. They are known to be correct. Why not put those chips into a big machine? You wouldn't need them in a data processing machine, but if you are selling a machine to engineers and scientists, it would be one board and it would have all those routines in there. And you'd be doing the processing in ROM instead of RAM – which goes twice as fast. I think you can buy a board; I've seen an ad in one of those magazines where a board has a microcomputer on it and then has all those routines on it, and the micro makes it possible to connect it to most anything. But still, as far as I know, Hewlett-Packard is the only one of the big manufacturers contemplating building those into a computer.

So I've been yelling about that for about a year now. If I can have them in my programmable calculus, then I can have them in my computer. Sure it takes more memory. If they decide to do it, there would be an extra charge on your hardware and you would have an extra board in your computer. I think that Hewlett-Packard is the one outfit that is doing it. And of course they do that small stuff; a lot of it does go to engineers.

Pantages: Which is the genesis, which is why I assumed it was that way.

Hopper: No, they're in the programmable calculator, not on the computer, which is silly. It is silly.

Pantages: It's another one of those things, like structured programming, that are supposedly a new technique.

Hopper: If they'd leave the top-down off and call it modular, I'd be perfectly happy. Because there are some programs you do not want to write top-down structures because all you do is generate a horrible amount of unnecessary code. And there are many cases where you should use a go-to instead of a perform. That is the trouble when you try to generalize anything. There are always specific cases in which you don't want the general case. And there are still uses for RPG.

What we tended to do in the programming area is when we got rid of the vacuum tube computers, we threw overboard everything we knew about programming and started all over again. And many of the concepts are now coming back again, just like the structured programming.

We had an infinite memory so we didn’t have to do any of those things anymore. So along with the vacuum-tube computers we threw away everything we had learned instead of carrying it with us. We wrote everything in modules from day one, independent modules, now that’s coming back again. The reason that didn’t get passed on is that those of us who learned it were all so busy writing programs, and we didn’t do the teaching. We left the teaching to some other people who didn’t realize it.

Pantages: You changed that.

Hopper: Tried to. There’s a good deal of divergence now between the academic computer scientist and the data processing people, and you’ll even find it throughout the industry. And I think we may get somewhere. The biggest fault of the computer scientist is that they don’t teach documentation. Once you’ve written a program and got it running, that’s considered it; you don’t have to report on it. And when I took a science in college, every time I performed an experiment, I had to write a report on it. Chemistry, physics. And they ought to do the same on computers. That’s an integral part of the experiment, but we’ll get around to it.

Pantages: I find you very accepting of human nature, but yet very impatient with it.

Hopper: It’s a combination of the two. You can’t help getting impatient, even though you know it’s not going to do any good. Like I said, there have to be a few people who get out in front and holler. And they have to learn not to be frustrated. Because getting frustrated only stops you. So you have to know that if you have a nice new idea it’s going to take five years to get it across. Then you don’t get frustrated. Every time you move an inch you feel happy about it. You just have to realize there is a world out there and this is the way it is.

I’ll never forget Burdette and Bloch wanted Commander Aiken to do something at some point and the Commander wouldn’t do it. And I had to try and tell Burdette that the Commander is exactly like the computer. He was wired a certain way and he’s not going to change that. And this is how you have to go at it and not get frustrated.

Pantages: What happened when you were pushing against operating systems and you were up there at the University of Pennsylvania?

Hopper: I didn’t really push against them. I was still at the stage of trying to figure out how I could get them apart.

Pantages: At this point you were at with the research group at the University of Pennsylvania, right?

Hopper: It wasn't until a few years after I got down here in the Navy that I began to really see what I could do with it. But the effort came from trying to get rid of all that general- purpose stuff in a program, in execution I mean. I began to look at the different parts of the operating system and see if they could be replaced by generators. That could generate the specific code for a specific problem. And then what would I have to know, and so on and so forth.

The generator of course comes from way back in the beginning, when I mentioned Betty Holberton's software generator. But what the generators really were was specialized compilers. They generated only a part of a program, or specialized program. They weren't general purpose. And actually, the first COBOL had an input-output generator in it as part of it. And then I realized that what we have now is a general-purpose interface to the operating system that did that function. All right, pull that piece out of the operating system and make it a generator again. And the same thing is true on the interrupts. You could for any particular program…it's a subset of the interrupt process that you want so you need a generator to generate that.

The concept of generator disappeared with those old machines, where we had to save space. I think it's time to bring it back again to save time. Once you get on to that you got on to the idea I was starting up this week that you can break up your compiler into separate generators and try to get rid of the operating system. It's going to take another five or 10 years, but we'll get there.

Pantages: What came out of your work at the University of Pennsylvania?

Hopper: Nothing that was really published except better COBOL compilers. Just a lot of thinking. A lot of people who moved ahead and got some new ideas.

1980: Back to the Navy

Teaching the Next Generation

Pantages: And then you said a couple of years after you got to the Navy…

Hopper: Norm Ream asked me to come back.

Pantages: You started to do more work in that area…

Hopper: No, the work was on the test routines, but I was thinking in the other directions. I haven't finished it yet.

Pantages: In terms of that thinking, are there specific projects ongoing within your group?

Hopper: No not yet. Still thinking about how you can do it. And doing a little piece here and there.

Pantages: Does the Center of Excellence have an opportunity for doing any work in that area or are they not staffed that way?

Hopper: Do you mean down the group in Norfolk? They are doing lots of thinking in all different kinds of directions. They're terrific. They're a bunch of youngsters. They don't know you can't do anything.

Pantages: Who's running it?

Hopper: Well, it reports back to Burdette. The guy down there in charge is Slater, who is in charge of building that micro. He's the one who built the computer aboard ship that they put in *Navy Times* and then the admiral wrote him the letter and that's how all that got started. The PR man for the ship took a picture of him and his computer and put it in *Navy Times*, and the admiral wrote and congratulated him, and he wrote a letter back to the admiral thanking him. But he went on for 10 pages single spaced and told the admiral what was wrong with the computers in the fleet. And he was absolute about what ought to be done about it, and he was absolutely right. And that's when the Admiral got him transferred from the Pacific Fleet to Norfolk and we started the Center of Excellence with him. Admiral told him to build a computer for the fleet. So he did. But Burdette had a hand in that all the way. Once Slater wrote back to the Admiral, and Burdette said let's get him.

Pantages: You made some comments to me and Burdette also mentioned young people and you made quite a number of efforts at training, particularly down there. And he said you are not as gung ho about the ability to turn them all into geniuses, as you were at the outset.

Hopper: I've always known you couldn't turn them all into geniuses, but you could upgrade all of them and some would be geniuses.

Pantages: What is your philosophy when it comes to training youths?

Hopper: The tremendous advantage of youngsters is that they will try anything. They have no prestige to put on the line. It's harder to get senior people to try something new. Because they have so much prestige on the line that if it fails you are losing. Youngsters have a tremendous edge there to try things for the first time. Their imaginations are more free. They haven't been told so many times they "can't do that."

Now they are growing up in schools where they had calculators and computers. Our senior people unfortunately have had piles of paper on their desk, big reports they had to read, things they had to decide. By and large they've had very little time to keep up with what's happening.

Think of the vice president in charge of information processing, and look at what he gets on his desk every day. And how much time does he have to read and study and hear about the newest things in electronics? Or an admiral or general or president of a company? So you are bound to get new ideas from youngsters, and you try to make use of it. That's true in any field. Some of them will continue to try new things, like George Baird and Dick Burdette, as long as they live.

Pantages: And you.

Hopper: Oh I think that's the most fun. I've always promised that on 1 January 2000, I'm going to call everybody up and say, "See, you underestimated." Which will be true. I think we consistently continually underestimate what we can do with computers if we really try.

Pantages: Where do you think they are going to do the most constructive things in the future, and where will computers be, potentially, most destructive?

Hopper: I'm hoping that the business and industrial and commercial companies all recognize what they can get out of their young people and give them a chance now and then. And you can see it happening because most of the new developments do promote economy and make more effective use of the computers. So I think it'll come in industry where there is an actual dollar value in doing things like that.

The ones that are building systems and computers today are not science and engineering people, but insurance companies and banks. So it will probably come from there.

Don't forget we have another generation coming that has grown up with computers; we can't even guess what's going to happen. There's a whole attitude of mind that is different.

Pantages: The only problem is that a lot of them can't speak and a lot can't spell.

Hopper: We'll have to teach them that then. At every college and university I speak, I get that every time, that they must learn to communicate. I tell it to the students, because if the students begin to get demanded, they'll get it. I think somewhere in the 60s, we forgot that people had to learn to communicate. We went off into relevance and forgot that communication was relevant. And that writing plain English was relevant. We've got to get that back again.

Pantages: As you have pointed out, that's the death of new ideas – the inability to communicate them.

Hopper: And the old guy in the back corner of the lab with his baling wire and string that puts stuff together won't get anywhere until they can tell people about it.

Pantages: Where is the biggest problem in the application of systems?

Hopper: We need more trained people, more systems designers who really know how to design a system. At the moment, it seems that NASA had the corner on them. And we need more programmers. Somebody says, "Well with all the compilers and everything, you're not going to need programmers." There is always going to be the next problem, and every time you finish anything, you see the problem that lay behind it. You always need people who can define problems, so we need the people that can look at the big picture and divide it up into the small pieces that can be coped with.

Pantages: What about the problem of ethics in computer use?

Hopper: That's the environment you grow up in, the learning of values. I think to a large extent, we abdicated the principles of leadership in the last 30 years, and we've got to bring it back again. We went overboard in the direction of managing everything. I pointed out particularly in the case of armed services that if you are leading a man into combat, you can't manage a man into combat.

I think the thing holds true also in business. We abandoned any concepts of the manager providing leadership, just management. I think we have to go back to something old-fashioned like leadership. A manager can't expect people to get into work on time unless he does. There's a little bit of setting an example in there.

Pantages: I think young people are really hurting for leadership.

Hopper: I think so, because I've watched them. They don't want total freedom; they want freedom within defined limits. They will test the limits; even the youngest youngster, if you tell him he can't do something, he'll try you out. When they find out where the limits are they'll go ahead and be free within those limits. There's no use trying to say that. At least not in the last 30 years. They wanted total freedom, which has brought many of them to a sad state. They needed some defined limits and standards.

Women's Liberation

Pantages: Do you think that women today are finally getting the shackle of liberation off their shoulders – I consider liberation to be a shackle today – and starting to figure out how to get ahead through their own wits and accomplishments?

Hopper: I would deplore some of the influence of women's lib in there. Some of them are still getting that. The sooner they get rid of that, the better off they will be. They asked me to be a chairman of the session, and I said, "Will I be the chairman?" They said, "Well, there'll be the program committee chairperson, and I said, "No way am I going to be a person. If I get to be the chairman, I will be the chairman." And I wouldn't go.

I've always been annoyed that they titter when they say, "She was the first man of the year," and I always break into the speech and say, "Man's a generic term." As soon as we stop wasting our time with a lot of silly nonsense, I came up long before those – any women's libber. What you do is do a good job and you get there. Now there is one handicap on women, is that they want everything. They are going to have some choices, they can't have everything. And women's lib is trying to encourage them that we can have everything.

Pantages: Have women gone as far as they should in the computer industry?

Hopper: I think they have done quite well. In the computer industry, they are way ahead of any other industry. I think they will go further as time goes on, not by women's lib, not by annoying everybody. But by doing a better job.

You really can't take six months off to have a baby to compete equally with the men who stay right there. That's one thing that's always going to be there. You have to make decisions. And I think we have gone amazingly far in the computer industry. We don't hear about everybody, don't hear about Betty Holberton. She did combine the two, she has two charming daughters and made tremendous contributions to the industry.

Pantages: She's retired now isn't she?

Hopper: Yes, I don't think she's ever been properly appreciated. You might do a piece on her some day, with all her contributions. They've never been fully recognized as I was. She wasn't as articulate as I was and she didn't stick her neck out as far. She made tremendous contributions right from C-10 code on.

Pantages: It does come down to selling yourself.

Hopper: You have to do that to get the money to do the things you want to do.

Of course, the universities frown on selling. So when you get trained in a university, you are not trained to sell your research, which you should be, maybe. A guy will publish a paper, and never talk about it, never sell it.

Pantages: You have been teaching for your entire career.

Hopper: Since 1931.

Pantages: In computer courses you've taught, how have the attitudes of the people changed?

Hopper: They've always been eager. The value to me is that I have to answer their questions, which calls to my mind something I haven't thought of yet. It's very good to be mixed with those students because they keep challenging me. If you teach a good course, the professor gets more out of it than the students do – because of their questions. They'll ask you about the newest things.

But there's a pre-selection there in that people that don't take the courses unless they are interested, so they are bound to challenge the professor. The same thing with the young sailors – they choose to go into computers. So they are bound to challenge things.

Pantages: Have you done much public speaking – that is, before non-DP people?

Hopper: No, only in so far that they get me before the Women in Government groups. And those have mostly been in government installations. But otherwise, it's ACM, DPMA, ASM, Armed Forces Communications and Electronics Association, American Association of Military Controllers. American Management Association. It's all been computers.

I spoke before Rotary Clubs a couple of times. But it was computer people who wanted me to talk to their pals. I've talked to practically all of the user's groups. Univac, the DEC users group DECUS, SHARE, and all the different user's groups for the different companies. It's all been computer people. I've talked for all of them. The only time I've been outside of that – and then I used a computer as an example of how you can move up with your job. Accent on learning. Accent on communications. It all carries over to any other job.

The Family Genealogy and Afghans

Pantages: What do you do in your free time?

Hopper: When mother died I found she'd been working on the family genealogy, so I decided I'd finish it. But then I discovered that your ancestors double with each generation, and I didn't go one line, I went all lines. And eventually I found myself back in the early days of this country reading Town Minutes to find out about people. And at that point I discovered a very interesting fact. I had learned in schools that the pilgrims landed in 1620 and that in 1773 we had the Boston Tea Party, and I knew nothing about the 150 years in between, during which time we formed all town meeting assemblies, the whole structure of government.

And I got started reading American history. I discovered that my brother-in-law descended from a bunch of Magna Carta barons, and so on, and I started reading English history. You ought to see the books around here on early American history and English history.

Then – on Christmas day every year after we've had Christmas dinner, my family tells me what they'd like the next year. Which means that there are several Afghan sweaters, Scandinavian patterned mittens, stuff like that that are on order. So I've had to make those. And there are various needlepoint things I just finished, a needlepoint rug for the bedroom for my sister's doll's house. I make things. I'm busy every minute. And then there's all the reading on computers.

I find American history entrancing, British history just as much fun, genealogy work I get to every time I get a chance to – and when I get to a town like Salt Lake City I always try to allow a little extra time to get to the libraries. When I get to Boston, I get to my favorite library, the New England Historical Genealogical Society.

I was born and brought up in New York City, and my mother's name was Van Horne, that's the old Dutch of New York City. So I get a whole slew of Dutchmen from her father. Her mother's name was Russell, and her great grandfather was Rear Admiral Alexander Wilson Russell. Which got me back to Pennsylvania, to the Scots of western Pennsylvania? And another branch comes from New Hampshire, Maine, all through New England. And after I got our family situated, my sister wanted me to look at her husband's, Billy Westcott coat of arms. They were among the original settlers of Rhode Island. He's a descendant of Ann Hutchison. It's absolutely fascinating. It's more fun than a detective story. You have to guess and look, guess and look. So that kind of research is fun, and I always figured I could have a second career, if I needed it, being a genealogist. There are a lot of things to do.

I've got four Afghans ready for Christmas, and several needlepoint things, and two baby sweaters and a baby afghan. See I can knit when I'm reading. So I'm very busy. Once in a while I take time off and read a detective story. Del Shannon. I used to read Agatha Christie. I never get to movies. I never get to television either. But lately I get to radio news, instead of television news. Radio gives you a lot more. Television can't tell you the news unless they have a picture for it. And they leave a lot out. We have one of those nice all news radio stations in Washington. I do read the papers. I have to keep up with what's going on.

Pantages: It certainly has been fun talking with you.

Hopper: I think all I've got is a lot of common sense and a willingness to work hard. Both of them are Scots and Dutch characteristics.

Pantages: A lot of people credit you with super intelligence, manifested in your ability to communicate with people in English.

Hopper: Don’t forget that I grew up in a day when there was no television. You read books. You acquired a vocabulary, which is a very useful thing. And then I have one other thing which is a help. That is the same as my sister and brother – we inherited the lean tough rugged physical makeup of our Scots ancestors. And that is a tremendous assist. Stamina and keep going. I haven’t had to make any major effort to stay skinny. I just naturally never did eat very much. I eat what I like. I never put on a large amount of weight. The same with my sister and brother. That’s an inheritance from our ancestors. We don’t catch things.

Pantages: Have you ever been sick?

Hopper: Well, I’ve been told that I had flu in 1918, not that I can remember most of it. I was only 12 years old. Never any long-term illness of any kind. It’s the same with my brother and sister. Since I’ve been down here since 1967, I’ve been down only once. The flu, I was down about a week. Compared to most people, I’ve been very lucky. I did have a black eye. But I don’t count that as an illness.

[*Tape stopped and was started again as she commented about World War II…*]

Hopper: The whole country was just doing one thing.

Pantages: People were all convinced then.

Hopper: If you go back and look at what was being fought for and what was happening in Germany. These days people have very little knowledge about what was happening and why we went in. We had a real purpose, a real end in view. We had to stop something, and we did.

Comments by Phil Vincent

[*This interview was on the end of the tape of the interview with Grace Hopper. The following is a brief conversation I had in 1980 with Phil Vincent, who was with Univac. I think he came from Remington and was largely involved as a manager in the customer service side.*]

Phil Vincent: They had this CAL, Computation Analysis Laboratory, over in Rich Avenue where people were really jammed in. And these were all desk workers. And the desks were piled in very close together with little aisles between them. And there were no individual offices. The second day I was there I went on a walkout in the section at lunch time. I saw two people in there and they were crawling around between the desks and every once in a while they’d rear up between the desks and each had water pistols and they were shooting at each other. And one of them was a guy named Harold Sweeney, who I understand died a while ago, and the other was Grace Hopper. You know, she was a little nuts like the rest of them. But she went in for this, she appreciated the light touch. But she was an extremely dedicated woman.

Pantages: She still is.

Vincent: There's a gal who would know Hopper from the woman's side, by the name of named Margery League. Now Marge is long gone from Eckert and Mauchly, her husband is a stockbroker. She was at Mauchly's funeral, and so she's obviously still in the area. She was down there in Hopper's day and can give you the woman's view of Hopper.

I can tell you a place where you can get them. One of them is Burdette Welch in Blue Bell – he's preparing a history of Univac. Underwritten by Sperry. There's another group and that's out of Washington. Otto Bernath is making preparations for a 30th anniversary for Univac.

I don't have any special anecdotes in relation to Hopper. For a woman of her brilliance she was surprisingly approachable. I haven't seen her for 10 years. I would like to see her again; I was genuinely fond of Hopper. I always felt that if you were assigned credit for the success of Univac, I would say that Mauchly was the indispensable guy. Eckert was a brilliant engineer but Mauchly was the guy with a vision. Now Hopper was the counterpart of Mauchly, and if you want to get a great view of Hopper because he knew her after the early days, talk to Neal Gorchow. He knew her starting in the 1950s. He would have the real grasp of the importance of the contribution she made, which I think was just immeasurable and it corresponds to Mauchly's contribution.

Both Mauchly and Hopper exhibited creativity. They saw the need for bringing this thing out of the vast wilderness of black magic and back into where it became a tool and didn't require a 20-year apprenticeship before you could use them. And that's what computers have become. People are selling them in stores and using them at home.

That's because of the base people like Hopper laid for making these things understandable. She simplified this tool in effect or the utilization of it so the average person could use that tool. I think that the fact that she had the vision to start the work that she did. Automatic programming was her group in the sense that programming up until that point was laborious – a one thing at a time. What she started was to use the computer to program itself. I think she was one of the true contributors who made giant steps forward. Without her, they wouldn't be selling these things inside the stores.

Pantages: She gives a lot of credit to Betty Holberton.

Vincent: Yes, but unless someone points them in the right direction they would never go down there, they wouldn't see the path. Hopper was one of those, this was an area in which you would do some work. She was one of the innovators.

Pantages: She didn't sit there and say "My God we're making history."

Vincent: That’s what I’m calling vision. That’s what Mauchly did. Originally they had a computer, a number-cruncher, but it took three days to load the numbers in. And what he did was convert it to where the number-cruncher was incidental, put lot of stuff in and a lot of stuff out. Now Hopper did the same thing. She made this applicable to business.