Oral-History:Betty Cooper

About Betty Cooper

Betty Cooper was born and raised in Raynes Park, England. She began working for J. Lyons & Co. where she was quite involved in both the LEO I and LEO II computer projects that would change the field of business and computing. As one of the first few women programmers of both computer projects, Cooper was a pioneer in the field of business computer programming. She also wrote the Braille program for the LEO projects.

In this interview, Cooper talks about growing up in Raynes Park, how she got involved in computing at J. Lyons & Co., and working on both the LEO I and LEO II projects. She also reflects on both of the projects and the status of women in computing today. Cooper also talks about her family life, completing programming working from home, and the progress of the field as a whole.

About the Interview

BETTY COOPER: An Interview Conducted by Janet Abbate, IEEE History Center, 14 September 2001

Interview # 585 for the IEEE History Center, The Institute of Electrical and Electronics Engineers, Inc.

Copyright Statement

It is recommended that this oral history be cited as follows:

Betty Cooper, an oral history conducted in 2001 by Janet Abbate, IEEE History Center, Hoboken, NJ, USA.

Interview

INTERVIEW: Betty Cooper
INTERVIEWER: Janet Abbate
DATE: 14 September 2001
PLACE: Windemere

Background and Education

Abbate:

This is an interview with Betty Cooper on September 14th, 2001.

To begin at the beginning, can you tell me when you were born and where you grew up?

Cooper:

When I was born: 1930.

Abbate:

And where did you live? Where did you grow up?

Cooper:

Raynes Park. It's a suburb of Wimbledon, which is on the opposite side to where the tennis is.

Abbate:

What did your parents do for a living?

Cooper:

My father was an accountant, but he didn't have any written qualifications. His parents were tailors. In fact, they were German tailors. They came over to London in the late 19th century, because there was work there and there wasn't in Germany.

My mother's parents: her grandparents were also tailors from Germany, but her mother—I'm not sure if she had a job, or whether she was just married. Her father was a hairdresser, a men's hairdresser.

Abbate:

Did your mother work?

Cooper:

She had secretarial training and worked as a typist for a solicitor firm. She was sent to Berlin in 1913 to perfect her German; there she worked as a nurse and remained in Berlin for the duration of the war. After returning she again worked as a typist until she married in 1923. But of course, once they were married, the rule was: women didn't work; they stayed at home. She actually had five children; the first one was still-born. They moved to Raynes Park, where they had a flat for a couple of years, I think, and then bought a house.

Abbate:

Where were you in the family?

Cooper:

I'm the third living one.

Abbate:

And you said you had one brother and two sisters?

Cooper:

One brother, and a sister older, and a sister younger.

Abbate:

What kind of schools did you go to?

Cooper:

An infants school—the equivalent of a primary school—and a junior school. I was eight and a half when the war started. I think that school was evacuated, so we went to another school nearer in to Wimbledon; it was similar. Then there was the eleven-plus exam, at that time. You can have two attempts: my sister passed the first time, and went to the local grammar school. I was a little bit younger than her when I took it. I passed the entrance exam, and went to the grammar school. Then I had a chance to take the exam for a second time in order to waive the fees, and I passed, which meant no more fees. [Cooper had commented before the interview that "in those days," most parents would only pay for their sons' schooling, not their daughters.']

Abbate:

So the eleven-plus is an exam you take to . . .

Cooper:

To go to grammar school; yes.

Abbate:

To go to grammar school without paying?

Cooper:

Yes.

Abbate:

And there was a separate entrance exam you could take, but then you'd have to pay the fee if you didn't pass the eleven-plus?

Cooper:

Yes.

Abbate:

Did you have an interest in math or science as a child?

Cooper:

I was always good at math. Not so good at written English, but I suppose I got by.

Abbate:

And did you have any plans for what you would do with this? Did you think you'd go on?

Cooper:

Well, not really. I could have gone on; I mean, if we could have afforded it, I could have gone on to university and taken a math degree. But we couldn't, so I decided—well, what else was there? There was accountancy, which you could learn on the job, and go to evening classes.

Now, my elder sister was a bit annoyed at not going to university. She got a job with a technical firm, and took a General Science degree by evening classes. My brother trained as an electrical engineer, in London. He was at university; but he got a scholarship, so that helped with the fees.

Abbate:

It sounds like your whole family got into technical fields.

Cooper:

Yes. Well, my younger sister was trained as a nurse, and she ended up as a District Nurse. She's now retired.

Abbate:

Did your parents encourage you to have . . . ?

Cooper:

They wanted us to have a good education. I suppose they thought that was the key to getting on and getting a good pay.

Abbate:

So you left school at age sixteen?

Cooper:

At a grammar school, you had to stay until you are at least sixteen; so I think I was sixteen and three-quarters. I had a year in the sixth form, which was on the way to the equivalent of A Levels. Again I took two extra O Level subjects: advanced mathematics and mechanics, in which I did very well.

Abbate:

How did you get your first job?

Cooper:

There was an education advice system, and you inquired what you could do with the subjects you were best in, and accountancy was one of them. With normal accountants, there was a fee attached to being trained. You had to pay a fee, and they would undertake to teach you accountancy, with minimum pay until qualified: this was for auditing, accounts, and that sort of work. With cost accounting, you started as a cost clerk, and you learnt the rudiments of costing, and if you wanted to go further, you took evening classes for exams that led to a qualification.

My first job, I had to learn about the printing trade and costing for printing, and at the end of two years I decided that I wanted to move on; it wasn't very well paid. So I looked around for adverts, applied to Lyons, and was accepted to work in their costing department.

Abbate:

The first job was in London?

Cooper:

They were in London, yes.

Abbate:

Did you move to London?

Cooper:

No, you get on the train and travel! It’s only about ten miles from our home in Raynes Park.

Abbate:

I forgot to ask: The grammar school you went to: was that mixed or was that all-girls?

Cooper:

All girls. My brother went to all boys. There were very few mixed at the time. Of course, now they’re mostly mixed.

Abbate:

So you started at Lyons in the early ‘50s, then?

Cooper:

1949, September.

Abbate:

And what was your position there?

Cooper:

I was employed as a labour cost clerk. The Costing section, where I worked, was part of the Statistical office, where costs were controlled and analyzed by product within the various food production factories (bakeries, ice cream, teashop savory meals, etc.). I produced standard labour costings for new products, with help from the Planning office (who provided an operation list) using standard factory cost rates. I also weekly summarized labour standard costs for items manufactured during the week, per factory department, and compared these with wages paid, together with expected indirect labour costs. I investigated any large discrepancies, referring to detailed wage dockets used to calculate wages (from factories). Our Supervisor would report to the appropriate Directors, who might have to take action to improve performance.

Abbate:

To spend less on wages?

Cooper:

Yes: to stop wasting time, scrapping products, etc. That was quite interesting.

Getting Involved in Computing at Lyons

Abbate:

At what point did you get involved in the computing effort at Lyons?

Cooper:

Well, whilst at Lyons . . . I completed the costing course I was taking for printing, and then decided to study for a Cost and Works Accountancy qualification with exams in stages. If I'd done this at evening classes, it would have taken a full five years; but if I did it by correspondence course, I could do it quicker, so I took a correspondence course. That's divided into various subjects, building up to an Intermediate Level, which I passed in December 1953 and continued to work for the final exams. Lyons were quite a progressive company for that time: if somebody acquired qualifications, they felt they ought to pay them more—but they won't do this unless they're making use of them, of course. As there was no advancement vacancy in the Statistical office, I was selected for an Appreciation Course with LEO.

Abbate:

An Appreciation Course?

Cooper:

Yes. They were looking around for various new potential programmers. Most of their programmers had a math degree or some other degree—thinking, I suppose, that if you worked up to a degree standard, you could be a very versatile employee and pick up new ideas very quickly—but they were also looking around the Lyons office teams to see if there were any potential candidates there that they could draw on. So I went on the Appreciation Course. It was a one-week course. It just covered the rudiments of binary arithmetic, simple programming, how the computer worked; and it took a week. It was quite an intensive week! [laughs.] I think they were trying to find out how you would do, and at the same time let you realize, "Would you like to do it? Were you interested?"

If you worked in the Statistical Office—we sort of pooh-poohed the idea of LEO. I thought, "Oh well, that's a silly idea. What can they do?" [laughs.]

Abbate:

Really!

Cooper:

Yes. I've written that down. [Reading from four printed pages entitled "Betty's Early Education, Training and Memories at LEO"]: "To use a computer to carry out office operations: it was new, and viewed with suspicion by many clerical workers. The concept of a programmer writing instructions to operate a sort of robot to carry out their work was difficult for them to visualize." That was up to the office management to sort out! [laughs.]

Abbate:

So you were skeptical when you started.

Cooper:

I was very skeptical. And I suppose because I was skeptical, I didn't worry about how I did—and so I did quite well!

Abbate:

But you thought that you might as well see what it was like?

Cooper:

Yes. Why not? [laughs.]

Abbate:

Was that the first time you had had any contact with a computer?

Cooper:

Yes. Well, we didn't have contact at that point. We knew it was there. Lyons had a personnel policy for all their higher-level management—potential management trainees—to have a tour around the offices and learn from people who were doing a job what they did; so I had had contact with management trainees, who might have been for LEO, to explain what labour costing is. But I hadn't seen the computer. It was a huge thing at the time, as you probably know!

Abbate:

So, they were teaching you . . .

Cooper:

They were testing you.

Abbate:

. . . sort of basic things about . . .

Cooper:

Binary arithmetic, what the computer did, showing you simple programming.

Abbate:

And that would have been in machine code at that point, the programming?

Cooper:

I think that they did give us some simple machine code, but it would also cover a little bit of flow-charting: how you were the tackle the job. Very simply, like "How would you program a computer to say 'X plus Y equals Z'?" I mean, that's a simple thing, isn't it?

Abbate:

So you'd work it out logically on the flow-chart, and think about how to translate it into [machine code]. What happened after the week?

Cooper:

I didn't hear anything for a while. I think they had to train somebody to take my over job before they'd let me go! [laughs.] I joined LEO in January 1955; I think they delayed it about a month while I trained somebody else.

Abbate:

So you'd taken the course at the end of '54.

Cooper:

The Appreciation Test: it was really a test, not a course. Yes, it was some time in the autumn.

Working on the LEO I project

Abbate:

When you joined LEO, how many people were already working on the team?

Cooper:

Well, there were about ten including one female programmer, Mary Blood. You've heard of her? Have you contacted her? I think she joined them in 1953.

Abbate:

What's her last name now?

Cooper:

Coombs.

Abbate:

Right! Right, right; I'm going to see her later in the month. Yes.

So you were the second woman on the LEO project?

Cooper:

Well, together with my future sister-in-law. We started the full Training Course about five weeks later, together with another person, a male. So I suppose you could say we were the second and third females.

Abbate:

What was her name, your sister-in-law?

Cooper:

Her name was Pat Cooper, but she's now Pat Fantl.

Abbate:

Oh, right! Okay.

Cooper:

So in the books, she is Pat Cooper, and I am Betty Newman.

Abbate:

All right: So you were the second and third . . .

Cooper:

Yes. It was quite an intensive course, the programming course.

Abbate:

Were you immediately interested in getting involved with the computer after you took the course?

Cooper:

Well, yes, it was something new! Now, what did we do? [Looking through her notes from the programming course.] We learned the basics . . .

Abbate:

So these are the actual course notes? You still have them?

Cooper:

These are the course notes that I took. The course covered all aspects surrounding the computer, briefly: its background, simple engineering circuits, how to do a job, programming actions required, flow charts, input/output, accuracy, checking, testing, setting up and starting a program on the computer (initial orders), original job specification, operating instructions, etc.; ending with writing, testing, and correcting a dummy program.

[Reading from notes:] "How did it start? You press the Start button, and it's got an initial order, which reads in the cards to set up the program. It reads in the program, which was on cards; sets them out; and then as soon as it's got to its end of cycle, it starts working with the program code." Of course, the program itself was written in machine code. There were three sorts of parts to the machine code, and it had to be compiled, so that it was tested that it translated so that it would work on the computer. I think—I'm not sure—it didn't have to be compiled like later programs; I think it was fairly simple. But [the program code] was really untested, and there's a program that read it in, tested it, and printed out some inconsistencies that it didn't like.

Abbate:

So before it was actually run, it was tested in some way?

Cooper:

Of course! We had to test things.

Do you want to go on and talk about the Training Course, or what we generally did?

Abbate:

Well, why don't we go on to what you actually did.

Cooper:

We would use the above procedure when programming after completing the course. You'd have a written specification, produced by either the Programming Manager or a Systems Analyst, and you'd read it, and then you'd ask questions, check out and tie up any ambiguities. You'd produce some flow charts. Now, if it was a simple job (one program only), you'd have between one and three input channels (cards and paper tape) and one or two output channels (punched cards and printer). You'd produce an input/output flow chart showing what data was input and what was output, then a detailed flow chart with square boxes for each simple operation and diamond ones for tests with two options—to change direction, also for impossible situations where a STOP may be necessary. The flow chart goes from the beginning to the end, and probably contains repeat loops where there is multiple data to be processed. And then, having got all the ends tied up, you could start coding. So you'd code it; you had to build—well, I'll come to that bit later. Having coded it, you got another colleague to desk-check it, to make sure that you had coded it correctly. "Have you left anything out that ought to be in?" and that sort of thing.

Abbate:

A desk check is someone reading it through, before you actually run it by machine?

Cooper:

Checking it; just checking it. They're checking it according to the flow chart and according to the job specification. Computer time was valuable, and obvious program errors had to be eliminated before trials.

Abbate:

How precise were the job specifications? Did they say, "We want to cost this job," or would they have it worked out . . . ?

Cooper:

Oh no, they'd go into a lot more detail. Yes. "What are the conditions of various things?" I mean, you had to cope with all the payroll. If it was a payroll, you've got to set up the basic data for each employee, which becomes "brought forward data." You have current data coming in, with "How many hours has he worked, at which rates?" (Some of them had more than one rate.) On the basic data, you have his date of birth; if he's taxed, his tax code; and anything that you want to know. And also on that record, you'd build up how much you'd paid him up to date, and how much you're paying him this week. And then, of course, you've got to allow for amendments to this, to correct the record if necessary, or if they change the rates. So you've got current data coming in, amendment data, and brought forward data (basic data and totals from the previous run); and you have to print a pay slip; you have to carry forward personal data for the following week's run; you want to carry forward some totals for the department; and you might want to carry forward some costing details.

Well, all those things went on one lot of cards, to be sorted them afterwards. But having got them, which order do you read them in, and what are you going to do with them? Well, that might be up to the programmer. But the systems analysis would tell you what information you have to hold for each employee, and what you have to achieve from that job.

Because of the memory size of the first LEO, which was 2K, it was quite a little tricky job to fit everything in! And it usually had to be done in a series of programs; but you've got to tie up all the programs together.

Abbate:

So the output for one program could be used in the next?

Cooper:

Yes, but you might find that you have to sort the cards—some go into one, some go into another—and each one would be attached to some brought-forward data for that. I mean, if you're doing the totals, you need the totals from previous runs, and you just have to tie up all the ends.

[Referring to "Betty's Early Education, Training and Memories at LEO":] I've got down here some problems and challenges, which I'll give to you later, if you want.

Abbate:

Oh, yes! Yes.

Cooper:

Then, having written it and checked it, it gets punched onto paper tape and compiled. The "compiling" program would read the written program instructions from paper tape (or for a program requiring amendments, cards and tape) into the computer and convert it to a readable "program running" format, which was punched onto the cards. Errors and inconsistencies found in the program would be printed out. These need to be corrected and the program recompiled. You need some test data, which is the next thing you produce. It has to cover all the different types of data that you're likely to encounter going into a proper program run. Now, if you were lucky, you were allowed to test the program yourself. But if you were unlucky, there was no day test time available on the computer, so you wrote out the instructions and the operators fitted it in when they could; and results came back to you, and they said it either worked or it hadn't worked. So it was up to you to check the results (printed and punched on cards) to find out what was wrong (if anything) and correct it, and put it back again for a further test. So it went on until you were satisfied the program was correct.

Abbate:

If it didn't run, what kind of feedback did you get?

Cooper:

Well, they always produced a computer log of what had happened, and where it stopped, and where it got unstuck, and you could try again, requesting a store dump of certain crucial parts. There were a lot of checks on it. (It's always possible that the first LEOs weren't as reliable as current programs, but I'll come to that in a minute.) Anyway, you got there in the end! [laughs.] You might have to have several attempts at reprogramming, and recompiling it, and putting it right, and trying again.

Then you'd have to write some operating instructions. You have to write or update the specifications, and when it is a live run on office data, you'd have to be ready to assist any queries that crop up. They might be program queries. They might be something that nobody's thought of, and then you might have to amend the program to cope with them.

Abbate:

When you're saying a "live run," you mean the actual people who want to use the data are running the program?

Cooper:

No, they would provide the data and LEO staff do the rest. [A "live run" was when LEO was being used in place of the manual office procedure.] They always had a parallel run to start with; so you compared the LEO results with the office results, and when everyone was happy with it, LEO took over running the job on a regular basis.

Well, that's the procedure that we followed. As I say, I think I've said that we had 2K storage, and that had to hold the program; the working storage, including the data as it was processed and totals accumulated; plus input areas and the output areas. There were also some fast calculation registers, for the arithmetic, and there was a console (which told operators how the program was running or completed or stopped). You did need quite a lot of ingenuity to fit your job into the computer!

Abbate:

To get everything to fit in that memory space?

Cooper:

Yes. LEO engineers had modified the original mathematical computers produced by Cambridge and various other places, conferring with the LEO management team and programmers. We needed fast input/output, because once in the computer, there wasn't a lot of calculation per item, or employee, or whatever data you were processing (store's item, if you're doing stock control, or personnel if it was labour costing), but there's a lot of repeat operations per data item. So they decided on three inputs: paper tape and two card readers; and two outputs: a card punch and a printer. They also introduced some buffers, so that the card reader can read at its own pace as soon as the buffer is empty; you haven't got to wait, in the computer, for it to read the card. So this speeded up the program running time.

Well, some of the challenges: I've listed some of the problems we had, the checks and challenges. As I mentioned, it was not a totally reliable computer; because it contained a lot of valves with a limited life and mercury delay tubes, a lot of things could go wrong; so the programs needed built-in checks for hardware as well as the software, to identify errors as they occurred.

Abbate:

How did you check the hardware?

Cooper:

Well, the engineers ran test programs. They had their own written test programs, and I believe Frank Land wrote some of them. Certainly the early programmers wrote them. These test programs were run regularly and also run before a live program. When I call it a "live program," it's an operating program doing the office work.

Now, the checks we had within the program would be on the data cards being read in. I'm not sure about the decimal cards; I believe they had an extra digit so that it had to add up across, but I'm not sure what amount. Paper tape had an extra digit. But the brought-forward and carry-forward cards were all in a binary format, so that instead of one column / one digit, you had something like twenty little compartments, it was binary format; and to make sure they were read correctly, there was always a check total card at the bottom, so that the first thing you'd do when a card is read is to check it: "Does it add up?" That always happened, and so you'd build a stop test into the program: "It has not read it properly; either reread it or abandon." It's up to you—or up to the operator—[whether to reread or abandon].

There were other checks. You would read in the data, and at the end of the run you would read in the totals for that department; so that at the end, you'd check that the total on the totals card is the same as all the brought-forwards added up. So that's another check. You'd also build a restart procedure into the program. You'd have to estimate how long the run is going to take, but about every ten minutes, you'd produce some restart cards: so that if you later found, "Well, it was all right there, but it's all wrong there at some later point," you could restart after the earlier point, and read in the appropriate restart cards you've produced, and carry on from that point.

Abbate:

So, restart cards would preserve the state of the memory at that point?

Cooper:

With the amount of information you've processed at that time, that is where it has got to, yes.

Abbate:

So you could load those back into the registers and carry on.

Cooper:

Yes. You'd restart that, yes. So we had restart cards—and reconciliations. Well, they usually had reconciliation programs, too; but within the original program, you'd have to check at the end of each department that the sum of brought-forward totals that you've individually read is the same as the total that you've brought forward—that it's correct. The program has to include instructions to do this.

We had to do some calculation as to how long the coding operations took for a normal straightforward cycle. Now, we had a list: we knew how long the "add" operation took; we knew how long the "subtract" took; we could calculate the multiplications and the tests. You wanted to get all the work done within the computer within the time it took for the longest input or output per data cycle—it was usually writing cards or printing that took the longest time. You wanted everything to work within that cycle. Otherwise, it's a double cycle and increases the time of the job—LEO operating time was in short supply—and of course it affected the cost! I think it was costing £50 an hour at the time, if my memory's correct, [so] if the job was taking half an hour, that's £25. Now, how much did it cost [to do it manually] in the clerical office? It is a very sharp calculation! The LEO operation must save money to be economic; otherwise it's not worth doing on the computer.

Abbate:

Now, let me see if I understand this: The actual time that the computer was executing instructions was supposed to be the same amount as reading in the data?

Cooper:

Well, probably writing.

Abbate:

Writing out the results?

Cooper:

Well, probably writing. It took longer to print a line than to read a card. If, for one employee, you've got to read in cards and write a card, or print: you know how long it takes to write a card; if you translate that into the time—I can't remember what they called the time; "micro-something"—but if the calculation time for an employee or data item was within the time, you were fine. If it went just over the top, it's two write or print cycles, not one!

Abbate:

Is it doing the calculations for one employee while it's writing results for another?

Cooper:

Yes, it probably would; output would be in the buffer. It could quickly put it in the buffer—but only if the buffer's empty!

Abbate:

Okay, so that way it's not using any more time than it has to. If it has a minimum amount of time to be writing out results, that's the time available for the instructions to run.

Cooper:

Yes. As soon as it's got the result ready, it moves it to the buffer and it will start on the next item. We might even have had two buffers; I can't remember. That buffer's waiting for the previous card to be punched, and as soon as it's punched, it takes [the data waiting in the buffer] and punches it, and leaves the buffer empty—ready for the next one. So, the cycle's got to be within that time, because if you miss it, it's two cycles! So that rather increases the time.

In our office we had a joke: We would go over the program and think, "Well, how can we save half an order (a programming order), just to save a little time?"

Abbate:

And how did you learn these kinds of tricks? Did all the programmers share tips?

Cooper:

Well, you discussed them! You discussed them with each other—and, of course, you learnt a lot of that on the training course.

Abbate:

That one-week course?

Cooper:

No, on the full 5-week training course. You had a model job at the end, which we had to run and get working.

Abbate:

That must have been some course! [laughs.]

Cooper:

It was! It was run by programming staff—LEO staff—and it really covered everything. The technical (hardware side) was kept to a basic minimum, but all the programming techniques, yes. And we had some exercises to do during the course, and we each were allotted a programmer who would answer our questions and help us along, This person was generally someone other than the giver of the talks, but it could have been the same person, because there were various different lecturers. There were only three of us on this course, you see, so we had plenty of individual attention.

Abbate:

Two women and a man.

Cooper:

Yes. The man was an Indian, and he decided that he didn't really want that job. He was a mathematician working in the bakery sales office, I think, and he decided—as he was doing an evening course—he didn't want the extra responsibility.

[short pause while Betty Cooper's husband enters the house]

Cooper:

The program log did help with debugging.

Of course, with the original LEO I and the first LEO II, there was no alphanumeric; it was all numeric input and output.

Abbate:

So how did you do alphabetical information?

Cooper:

I think it was coded. I think the employees each were given a number.

Abbate:

Ah.

Cooper:

We probably didn't have their names and addresses . . . Again, I can't remember a lot of these things; it's over fifty years! No, not quite fifty years: forty-five years. It's a long time! [laughs.]

There might be other checks we did, which I haven't written down. [Refers to notes.] I've been through the normal procedure . . . I can tell you which jobs I had a hand in.

Abbate:

Yes, what were some of the main projects?

Cooper:

Well, we started with payroll. Pat Cooper and I worked under Mary Blood. She was a very dominating person, I didn't find working with her very easy! [laughs.] So I transferred to John Grover's group, which was covering costing and stock-control type jobs and statistical, until he left; and then Frank Land took over from him.

Abbate:

Was the idea to go through Lyons and automate one office after the other?

Cooper:

Well, the jobs that we did came from our Programming Manager. I think he started in Systems Research, and he talked to [J. R. M.] Simmons, who was the Director in charge of office management, and they produced the jobs. There were some jobs for outside firms as well.

I helped with the bakery evaluation program. This was one of the first ones they did. Now, I knew that from the Statistical Office side of it: it was valuing sales, and splitting each operating cost into material, labour, and overheads costs, a run carried out weekly. Long before that, before it was done there, it was done in the Statistical Office, so I knew both sides.

Abbate:

So you had done it manually, as well as on the computer.

Cooper:

Yes. But only amendments to the program. We might be involved in amending it, I think, more than anything else: any extra facilities they needed to put in it.

Abbate:

Did automating the procedure change the kind of information they had available?

Cooper:

It could produce extra information quickly. Totals, which would be another job in the office: it could automatically print them out at the end of the run!

Costing: I costed—maybe not early on—a job for a subsidiary of Lyons, for clock production. You might have read that in one of these books, or some way; I don't know.

Abbate:

No, I haven't.

Cooper:

It was quite a complex organization, because clocks have parts; subassemblies; more subassemblies. Now, parts could be used in one or more subassemblies; subassemblies could be used in one or more type of clock; and finally you've got the clock. Now, the program—or the set of programs—had to organize it to break down the clocks into parts, so that you could produce production lists according to how many you wanted to make. Then you had built up the costs, so you know how much they cost, [and] you might have to revise the prices. I can't remember now how we do it. I know it was a complex program! [laughs.]

Abbate:

Why did a catering company have a clock subsidiary?

Cooper:

I don't know! It was a Jewish family, and the family had aunts and uncles and, you know, it was a wide firm. So it wasn't Lyons, but this Director was had interests in various other products. I mean, we did jobs for other people who came and asked. Later on, the fur trade wanted something doing to produce some costs for getting in the skins and producing things from it afterwards. I can't remember now what we did with that, either, but I do remember doing something! [laughs.]

Working on the LEO II project

Abbate:

Did you feel you were on the cutting edge of business practice?

Cooper:

It was always interesting, because as soon as you finished a job, you're starting a new one. We had a lot of job satisfaction. You know, "We can do this." Where it would lead to, I don’t think any of us knew; but they started working on LEO II—which was an improved model and faster than LEO I—fairly early on, soon after I joined them; and that cut down the time. In fact, all the machine coding changed, so that where you had—let me think—14 was “add” before, and 3 was “subtract” . . . It changed the numbering system, so you had to know two number systems! [laughs.]

Abbate:

Oh, because you were using them both.

Cooper:

Yes. You were [still] doing things on the original—well, you weren’t coding new ones, but you might be amending and keeping them working. And you had to re-code those onto the new system, so you were translating; rewriting.

Abbate:

That must have been confusing.

Cooper:

Well, it’s amazing; it sort of fell into place. You soon could translate easily to the new code.

I think the size of the numbers held in the computer was increased, as well. It worked with a decimal point, I think. I can’t remember now. How it held the numbers: I think there was a check digit on that, to make sure that it hadn’t done something to the number within the computer, and that was all part of the electronics. Whereas a word could be held in 34 binary digits in LEO I, it went up to 38 binary digits in LEO II. And you were working in binary as well as working in decimal. You could translate. We’ve got a translation, I think. [Looking through papers:] I’ve still got my original card! [It shows] what the number would look like.

Abbate:

Ah, so this would translate from binary to decimal.

Cooper:

Yes. What is the binary power, you see? And the power would be one, naught, nought, nought, nought, nought, nought, nought [10000000], or something, depending on how many it was. We had to translate the data. You had to sort out and make sure the data produced was correct. Your trial data—which you tried to keep simple as possible, but it had to cover every operation, every different type of data.

Abbate:

Did you find that this drew on your mathematical aptitude?

Cooper:

I think so. Logic: it needed somebody to think logically. Maths is a lot of logic.

It always amazes me, at the moment, how mathematical operating a computer is now—whereas maths at school was always, “Oh, I can’t do maths!” computing’s different.

Abbate:

You mean for people nowadays?

Cooper:

Yes.

Abbate:

Interesting.

Cooper:

But at the time, you had to think mathematically. There weren't very many who came up from Lyons to LEO. Leo Fantl did, he was one of the first programmers. (He married Pat, later.) Who else was? John Grover was another. But most of the others had degrees. Leo worked for his degree at evening classes; he came from Czechoslovakia, at the beginning of the war. But his history's in there anyway.

Abbate:

So they were coming in from outside?

Cooper:

Leo Fantl worked in the Lyons Planning Office before joining the team.

Abbate:

Were you training new programmers as they came in?

Cooper:

All the outside degree people went on the fall training course.

Abbate:

So that was held periodically?

Cooper:

Well yes, yes. It might be every couple of months, as much as that. As I say, we got the normal programming procedure instilled into us, and we kept to it—kept us working efficiently and [using] good practice. [It was] worked out by somebody.

Reflections on the LEO I and II projects

It was an interesting life. It was more interesting than doing the same office job week in, week out! Even though I found that labour costing was quite interesting at the time, because there were always new jobs, and new items to be costed or changed or what have you. But the idea was to take all the slog out of the office work, which I think on the whole is probably a good thing!

Abbate:

Did it also increase accuracy?

Cooper:

Well, yes.

Abbate:

Once the program was right!

Cooper:

Certainly, if things went wrong, they went madly wrong! [laughs.] So you knew. And of course if it goes right, it doesn't make a mistake; whereas in the office, they could.

So what else did I do? A stock-control program. It was a very good one, because as well as keeping track of the stocks and the prices, and so the value of the stock, you can hold reordering levels and usage; it can tell you, "Your stock's gone down to your reordering level. Make some more, or buy some more." So, we had reports on that sort of thing. It's a very good example.

The bakery evaluation job was a sort of an extension to that. Bakery evaluation would be to] keep control of the stocks that you're reordering. Mostly it was left to the departments to do the ordering, but to have some prompts was a good thing.

There was the tea shops job, of course, but I wasn't involved with that. I'm not sure who was head of that one.

Abbate:

The accounting for the tea shops?

Cooper:

Well, no. In the various tea shops, as the weather changed, they wanted different products delivered. So they could wait until three or four o'clock in the afternoon to phone up and say, "I want this instead of that," and they'd get it delivered first thing the next morning. So they could change at a late time, and that was very good, because the managers of the tea shops did not have to juggle the figures themselves. They could concentrate on managing their staff instead! It's the fast reaction [of the computerized system]: once it's working, it does very well. If you get figures quicker, you can take decisions more quickly.

Abbate:

But you were doing more the head-office kind of automation, as opposed to at the tea shops?

Cooper:

Well, the tea shops was all part of the Lyons empire, so it did come in. We did all the payrolls. They started with the payroll of the bakery office, the bakery production; and when they were happy with that, they extended it to everybody, including tea shops and corner houses (a larger type of tea shop) and all the different production departments: tea, ice cream—they all had their little hierarchy. It just got extended. All we needed was more computer time, to fit everything in.

Abbate:

You talked about the economics of doing the computing. Was it less expensive to do it on the computer than by hand?

Cooper:

Oh, yes! They wouldn't put it on otherwise; they wouldn't do it. We had to work it out, yes. It had to be worked out.

Abbate:

So even right from the beginning . . .

Cooper:

Yes. I remember, when Eveready—you've heard of Eveready torches and batteries, haven't you?

Abbate:

Yes, I have.

Cooper:

It was just before I left them, I think; around about 1959, 1958. They had a man over there who went into all the details: "How much were the data preparation costs? How much for this? How many staff can he save in the office?" And he worked it out very carefully, and [asked], "How much would LEO charge you to do the job?" And if it wasn't economic, we didn't do it!

One of my best friends worked for Eveready , and she came and went on the training course, and she worked under me for a while, doing Eveready programming.

Abbate:

Did you become a supervisor of programmers?

Cooper:

Well, I only had a small group, I suppose, for about the last year I was there. I left in 1959. And I had my daughter Shirley later that year.

Abbate:

Did you meet your husband at Lyons?

Cooper:

No, I met him through his sister [Pat Cooper], who did the same programming course as me.

Abbate:

Oh, I see.

Cooper:

They shared a flat in London. They came from Bradford, in Yorkshire.

Abbate:

Did Lyons show a lot of appreciation for the computing group? Was there a sense that you were very important to the company?

Cooper:

I think the management did. But I'm not quite sure about the actual lower people in the staff there. They were all very skeptical. Well, I was skeptical when I first joined them! [laughs.]

Abbate:

And did you actually work with the people whose work you were automating, or was that just done by the Systems Analyst? Did you interact with them?

Cooper:

You did have some contact, but probably not a lot. Yes, if you had some queries about “How have we got to cover this case, or that case?”, you would probably [talk to them]—not the individual people, but their supervisors. The engineers and the programmers talked together, because they would introduce new things, new facilities. They were forever updating the facilities on the new computer. When they started building computers for other firms—which they did—they had to make it alphanumeric. The first LEO II wasn’t, but the second one was. And then, I think, they had to introduce new things. Potential customers would say, “We wouldn’t buy it unless we can have so-and-so,” you see; so extras were introduced.

Abbate:

Did they consult with the programmers, in terms of what features the machine would need?

Cooper:

Yes.

Abbate:

So there was sort of a back-and-forth.

Cooper:

Well, yes, they were always talking. They were in close contact with the programmers, and if they wanted more tests we would provide them. I did write a test program—well, not a test program: a program that kept details of the computer valves and their life span. The engineers would provide details about which valves were in the computer at this time, and how long they were in use; the program had to calculate and report on valves nearing the end of their life—i.e., which were likely to go wrong. So if they had problems, they’d check those first. I did write a program to update this information.

Abbate:

So you had to keep track of the hardware, in a sense.

Cooper:

Well, I didn’t; the program did!

Abbate:

But I mean, as programmers, you had to be aware of which things were likely to fail.

Cooper:

Well, they would come to ask; they’d say, “Look, we’ve got this problem. What can you do?” So I think Frank Land sorted that one out, and I did the programming. Yes, they would come with the problems, and we would write programs for them.

Abbate:

For the engineers?

Cooper:

Yes.

Reflections on Women Programmers on the LEO I and II Projects

Abbate:

By the time you left in 1959, what percentage of the programmers were women at that point?

Cooper:

There were quite a few more. It was a little bit confused, because we had a lot of programmers who worked for outside firms, and they were coming in to learn it, because their firms were interested in buying a computer, so they needed their own programmers. When I started, there were about eleven programmers, and there were just the three of us women. I would think they were mostly male. A quarter might have been women, but no more. It might have been less.

Abbate:

So about the same as when you started.

Cooper:

Yes.

I don't think they differentiated between—I mean, if a woman applied and they were good enough, they came.

Abbate:

So you think there was just less interest from women in doing it?

Cooper:

I don't know. It's difficult to tell. Pat had a maths degree, but she did say at the time that there wasn't a lot of work available for women using maths. I think she worked for an aircraft firm, doing various mathematical calculations; and she did say it was difficult to get another job, because they a prospective employer in the same industry would apply to the firm she was working with, who would reply, "Oh, we can't let her go!" [laughs.]

Abbate:

Because they wanted to keep her. [laughs.]

Cooper:

She was not to join an opposition firm. But LEO was completely separate, so she was allowed—she could manage to move.

Abbate:

That's interesting.

Cooper:

She only stayed a couple of years, I think. She got married and her husband worked for Bell Labs in America, and she went with him and obtained a job with IBM writing in FORTRAN.

Writing the Braille program for LEO

Abbate:

Did you go back to computing after you left, at all?

Cooper:

I did some part-time work at home, writing programs for LEO.

Abbate:

This was in the ‘60s?

Cooper:

Yes. [Referring to notes on “Betty’s Early Education . . .”] I had a reason for writing this out, because my daughter’s been asking me about it. Between ‘61 and, say, ‘65, I was doing odd jobs for—amending programs for—LEO I or LEO II. I didn’t learn the language for the LEO III; that’s when CLEO came in, and I didn’t do anything with that. But I did then go on a course in 1967 to learn User Code for System 4, and after that I wrote a Braille program. That’s to say, you take the normal script and print it off in raised dots, in Braille format. That was a quite interesting translating program.

Abbate:

I haven’t heard of that.

Cooper:

Operators turned the paper upside-down, removed the printer ink and printing had a very sharp contact with the rollers to raise dots on the paper. A blind person could feel and read the document printed.

Abbate:

So this would just turn any text into Braille?

Cooper:

Yes.

Abbate:

I hadn’t heard of that. Who was that for?

Cooper:

I think it was the Civil Service. They employed Braille programmers. I couldn’t be certain, I know it was operational. I had a phone call after we’d moved up to the Midlands (my husband’s firm moved): “Can you come up and do it?” Well, I found it very difficult with children here, because that was in London—somewhere out in the suburbs—so I turned that one down. But the program was very appreciated.

Abbate:

So people were programming in Braille.

Cooper:

I don’t know what they were doing with it; I couldn’t tell you. Frank Land might be able to tell you about that. I don’t know.

Who else did I work for? That was Doug Comish, I think, at the time.

Abbate:

That was who?

Cooper:

Doug Comish. He’s also mentioned in there.

But that was when I wrote programs completely at home, and they were tested in London. I think I went up for some testing.

Abbate:

So it was fairly easy to work—to do this sort of program—at home?

Cooper:

It’s easy to program at home. All you need is a lot of programming sheets and the specifications, and you just get on with it. In fact, it’s quicker to do it at home than it was in the office, because you don’t have any interruptions!

Abbate:

Well, except your children, I suppose!

Cooper:

Oh, well, you don’t do it when they’re around! [laughs.] If they’re at nursery school, or if they’re at school, that’s okay.

Details on the Programming Sheets

Abbate:

What did the programming sheet look like?

Cooper:

[Looking through papers:] I don’t think I’ve got any. [Beginning to draw:] Well, it was just . . . I think we had . . .

[Draws something like this, where the x’s represent spaces for 0’s and 1’s to be filled in:]

No. on coding sheet Action Register or location specifier Size of number (0 for normal size, 1 for double size) Description written for programmer’s use xx xx xxxxxxxxxx x

Abbate:

It’s sort of a grid.

Cooper:

There were four columns. This [second column] was the action, which was a number. I think 14 was "add"; 5 was "subtract." We had no division; we had to use a mathematical subroutine. There was "multiply" or something; and "goto," where there was a test. Well, the action number went in there.

Abbate:

Okay, so that indicates what instruction you want.

Cooper:

Yes. There was a very small one—was it 01 or 10?—which told you whether, depending on the size of the number you were working on . . .

I've probably got it in here.

Abbate:

So that was sort of a check?

Cooper:

Here we are. [Pulls out coding sheet.] It would look something like that. So, that's the action. And that's the number of the coding [in the first column].

Really, it looked something like this: This last one was the size. I think at this side it numbered them down—the number of the order; we called them an "order." And then this one will carry on. And this [third column] was "Where do you find it? What are you working on?" You're working on this number.

Abbate:

Right: what memory location is it in.

Cooper:

Yes. And you had a separate arithmetic register. So this would say, "Add so much to the register." This would say, "Subtract so much." This was "Multiply the register by this." "Put it away; where are you going to put it." So they're very simple operations; but you had to break everything down simply that way.

Abbate:

But you had these preprinted sheets so that you could conveniently . . .

Cooper:

Oh, yes! Yes, you had a number of sheets.

Abbate:

Each line would have the action . . .

Cooper:

It could be that you only had the one going down, and this was—you wrote at the side what you were doing. Let me see what I've got [in my notes. Reading orders:]. "Replace the contents of the accumulator by the contents of the location specifier. Add the contents of the location specifier to the accumulator." The accumulator would cooperate with the register. "Transfer contents to somewhere-or-other, and clear the accumulator afterwards." "Transfer it and leave the accumulator as it was." That sort of thing.

Abbate:

And this last column is a checksum kind of thing?

Cooper:

No, I think it was the size: are you working on a small number or a double number?

Abbate:

Oh, I see! Okay.

Cooper:

Yes. I think it was just a nought or a one, and if it was a one, it was a double-size; if it was a nought, it was normal size. And of course some of them didn't use that; it was only if it was a number that it applied.

Abbate:

So it would know to read in twice as much data?

Cooper:

Yes.

[Reading:] "Programming rules: Write down the basic instructions. Test for END. Step on count." Every time it had done one order, it automatically went to the next one—but if you did a test, it tells you where to go to. Such as this one, saying "Go back to there." And you could write in this number, plus five, or whatever.

Here's another example. That's what the sheets looked like. They might have got it going down there, and going down there; but you would need room to write down what you're doing.

Abbate:

So you could do it on a blank piece of paper, but it was just more orderly to have a nice sheet where things are arranged ?

Cooper:

Well, you needed a sheet, because these had the order of sheet numbers printed on them; otherwise the data preparation girls wouldn't be able to punch the coding. They put it on tape, and it got translated into cards by the compiler program.

There's a sheet for you! It looked like that. You see, you say what you're doing; that's the coding. This is what you're wanting to do; and this is what it is actually doing; and then you go from there up to there. We had lots of these printed.

[Reading from notes:] "Accuracy." We had to work out how much you lose, because you're doing multiplying numbers and putting it back. I don't know that I understood the accuracy that much. Computational accuracy: how to work out percentages.

Abbate:

You were losing accuracy because you were rounding it off?

Cooper:

Yes: round-offs.

[Reading from notes:] “Post-mortems.” Post-mortem is: “Has it worked? If it’s gone wrong, what have you done about it?”

They printed these. They had a printing department, obviously, because they printed these sheets that we were writing on.

Problems with LEO, Circuits

Abbate:

Do you have a log where you had to log problems that you’d have with the machine?

Cooper:

Oh, the computer has a log, yes. It prints it up, and it was a great help in finding errors, if there are any.

[Looking at notes:] We’ve got “Simple Circuitry” that we had a talk on. They printed that, and we wrote down the notes: wavelengths and times.

Abbate:

Was that helpful?

Cooper:

Yes, I think it filled in background. [Looking at circuit drawing:] We called these “flip-flops.”

Abbate:

Right, the circuits.

Cooper:

The circuits.

“Timing”—ah, yes. We certainly had to learn about timing.

Where it was in the store . . . It’s a long time since I looked at these things. I’m amazed I’ve still got it!

“How LEO is set to do a job.” This was all the five-week course.

A flow chart: here are some flow charts. I think later the test boxes became diamond shape, so you could see the test boxes. You see, that’s a test, so you’ve got a tick and a cross: you either go that way or that way. [Looking at another symbol on the chart:] That’s to re-read the card, I think. You had to do this [flow-charting] for the whole job.

What’s this? This is the input-output chart: what goes in and what goes out.

When you’d done the course, of course you had something to refer to, but it’s only when actually programming that it really registers and stays there.

Abbate:

And then maybe you would look back and say, “Ah! That’s what this meant!”

Cooper:

And problems, you would discuss with other programmers—probably at coffee time or such like.

Abbate:

Did you document the programs?

Cooper:

You mean, write a specification afterwards? Is that what you mean?

Abbate:

Right.

Cooper:

Yes, we had to do that. We had to write out good operating instructions: what to do if it stops; what might be the reason. Each stoppage had a different number, so that we could then turn up and say, "Oh yes; that's what's gone wrong."

Abbate:

So when you say it had a number: it would produce a number on the error log or something? Or output some kind of signal?

Cooper:

Yes. Well, it would—If it stops, you can see it on the console. It had—I'm not very good on these terms, remembering them—it was like a screen.

Abbate:

Right. And it would display a number from where it had stopped?

Cooper:

Yes: why it stopped. So then they turn to the operating instructions: "It hasn't read the card right. Try re-reading it; put it back in."

Abbate:

Who actually ran the programs, once they were finished?

Cooper:

There was an Operating section who organized and ran jobs on the computer. Now, when you say, "Who is involved in programming?": There were also the Data Preparation section—the girls who punched the data on paper tape and verified it, also our programs and amendments; they were all girls, and they had a supervisor. And there were two others, I think: a plug board had to be fitted into the printer, where you specified how it's going to print, and they set up these print circuits to plug into the printer.

Abbate:

There was a different group of people doing the plug boards?

Cooper:

There were two that were doing that.

Abbate:

So the people that—the operations group, or the operators actually running the programs . . .

Cooper:

There were some female operators as well. Operators only ran jobs.

Abbate:

And were you in close contact with them? And did they come to you and say, “Oh this doesn’t seem to be working?”

Cooper:

Oh, yes! We were all in close contact with everybody who worked there.

Abbate:

Were you all physically in the same place?

Cooper:

No. They were in the operating room, mostly, and it had to be a fairly air-tight—no dust around, because that could affect the working [of the computer].

Abbate:

Did any of the Data Preparation girls ever move up to become programmers?

Cooper:

I think they did. If they were interested, they would then progress to do these plug boards for the printer, and I think at one of our reunions I went to, I found that those girls had gone on to programming. So yes, they did move up! They weren’t programming while I was there, but they were later.

[Looking at more notes:] That was our model job [from the training course]. We had to write it, afterwards.

Abbate:

Did you go back to LEO full-time at any point? Or were you just working temporarily?

Cooper:

I didn’t go back to them, because in 1969, we moved up to the Midlands, near Burton-on-Trent. You might have heard of Burton-on-Trent? North, near Derby. I did get a job in Derby in 1975, working for . . . They produced presses, and machine parts for other presses. I think originally that was an American firm: EW Bliss.

Abbate:

I’m not sure. So you were working for them full-time at that point?

Cooper:

Yes. More or less full-time, yes.

Family Life

Abbate:

So I guess your children were all in school by that time?

Cooper:

They were at school, yes.

Abbate:

You have four children?

Cooper:

Two.

Abbate:

Oh, your parents had four children; you have two.

Cooper:

Yes, that's right. Two's enough! [laughs.]

Abbate:

And what did they end up doing?

Cooper:

Jeff is an electronic engineer. He's working—I think he's programming, progress-chasing for meters, to read them automatically. But he's working in the electronic field, with programming, and of course he's got his little group who work for him.

Shirley, as I say, has a physics degree, and she was doing research for the Open University: theoretical nuclear physics. She did that for about ten years or more. Now she gives lectures for a firm called Connect, who retrain ex-programmers without jobs for programming on the Internet.

Abbate:

So they're going from COBOL to Java, or something like that?

Cooper:

Java—and CC+, is it?

Abbate:

C++?

Cooper:

C++. Yes.

Abbate:

Do you think your children—did they sort of follow in your footsteps?

Cooper:

Well, it must have had . . . They didn't follow in my footsteps, no; but they are doing the same sort of work, and they know—Shirley certainly knows a lot more than I do about it; I think Geoff does too! [laughs.] But then, the trend is computing, isn't it? Shirley taught herself C++ and Java when she was between jobs, she had been teaching at Open

University summer schools before that, so she's had some teaching experience, and I think Connect gave her a trial for a year. But I think the firm works on contract, so she's still working under contract; but it gives her some experience. She can go on somewhere else after.

Abbate:

Yes, I'm sure.

Cooper:

It's the job she likes.

Working for Bliss Company

Abbate:

Now, let's see: we had left you working for the Bliss Company, I think?

Cooper:

Yes. Doing all sorts of computer jobs. There were only a couple of us there when I started, and then he left and they got another one, and then another couple. But, we were doing everything on the 2901. Now, who would that be? ICL. It might have started off as a 1901 or a 1903, I can't remember, but it was ICL. It was a smallish computer, but it had its own computer room at the time. We were doing all sorts, like payroll, and stock control, and costing. You needed to be able to read—it was in COBOL, and they also had a program which we called FIND: you fed in parameters for a file (sort of database), and you got the results printed as you specified. It was very simple to read in a file that you'd created somewhere else and produce statistics and various other things. But COBOL, yes: a number of their programs were written and corrected, and I dabbled in all of them. You know, I had to read them, and find out what they did and how you could amend them; and if you weren't careful, you amended something that you didn't see a reason for it, and found out afterwards that there was a reason for it! [laughs.] I think one costing program, I found that they hadn't written the last record of the run at all, so they'd lost it! [laughs.] For years they hadn't recorded the last costing item that they'd dealt with. I thought, "I'm sure this can't be right!" [laughs.] So you would go around asking, "What happened to this record? Have you got a mention of it anywhere?" And we ended up by—they had the disk computer extensions connected to the computer; they weren't PCs, and you could only operate them to find information from a file on the computer. You could type in the number, say, for the part in the Sales or the Program Production Department, to find out how much you've got in stock and that sort of thing.

Abbate:

These were terminals?

Cooper:

Terminals. That's it; yes.

Abbate:

So they could interact with it.

Cooper:

Yes. But we had to write the programs for that. That was quite an interesting objective, because you could display data information as just a straight-line list such as you would print, but that's not easy to visualize. You really want to type in a stock item number, look down and see, "When did we last use it? How much does it cost? How many have we got in stock? Should we reorder them? How often do we use them?" And that sort of thing. They had all that at their fingertips, and you only had to type in the number. So I quite enjoyed writing that program!

Abbate:

Were you the senior programmer at that point?

Cooper:

I did end up the senior programmer, yes. [laughs.] It was a small, contracting size of firm. I mean, when they started up, they had 400 employees—that's not very big, admittedly, but I think they ended up with a hundred when I left! They weren't doing very well. They did most of their business on providing spare parts for these great big presses.

Abbate:

Printing presses?

Cooper:

They weren't printing presses, no. I think they were mechanical presses. Like producing, say, saucepans.

Abbate:

Oh, these were industrial.

Cooper:

Yes, industrial.

Using COBOL

Abbate:

And how did you like using COBOL?

Cooper:

I found it easy! Compared with User code, it's much easier! [laughs.] User code, or what is it? System 350. Are you familiar with it?

Abbate:

The IBM system.

Cooper:

IBM, yes. They were interactive. They were the same sort of thing. But COBOL was easier! But it's amazing what you could do with COBOL. The only thing you couldn't do was some mathematical calculations.

Abbate:

Could you not do them at all, or was it just too difficult?

Cooper:

It was the complicated formulae that you couldn't do. I mean, you could add and subtract and all the rest of that clerical work, yes.

And of course, my cost-accounting came in very useful. So I knew—I mean, I heard arguments: I'd hear two people talking about some system of what you can do with costing, and can you do this, that and the other? They'd turn to me and say, "Can we do that?" "No!" [laughs.]

Abbate:

How long were you at EW Bliss?

Cooper:

About ten and a half years! It was a nice place to work.

Abbate:

Why did you quit?

Cooper:

Oh, we moved here.

Abbate:

Oh, so you retired.

Cooper:

I was only 55 when I retired, but my husband Laurie was —I think he took an early retirement. He did research work for the Coal Board, and they were contracting and wanting [to reduce staff], so he got a "silver handshake." That's when we decided we didn't want to live in the Midlands forever. Would we move back South? Well, houses cost too much. Where else would we like to live? And we found this house [in Windemere], a nice place to live. Laurie doesn't like it when it rains and rains, but there are so many nice days in between! We do our chores when it rains.

[Tea break. Recording pauses.]

Reflections on Working with Computers

Abbate:

Just some general questions: What did you find the most satisfying part of working with computers?

Cooper:

I think working out the new methods of organizing it on the computer.

Abbate:

To take a problem and figure out how it could be done?

Cooper:

Yes.

Abbate:

Is there some particular thing you're most proud of, in terms of things you've worked on?

Cooper:

They were all quite interesting. I think that the new programs.... When I was working at Bliss, we were translating or writing programs for a subsidiary of theirs in Scotland, and their accountants said to me, "Don't you like carrying on with this subject afterwards?" And I said, "No, I like to start a new one!" [laughs.] You've told me what you want it to do, so let them get on with it and run it. I mean, he's just using it, as you would in an office.

Abbate:

But there was always the chance of something new?

Cooper:

Yes!

Abbate:

Did you ever feel that as a woman you weren't—you didn't have access to promotions, or pay . . .

Cooper:

Well, I wasn't really looking for promotion, once I was there. I mean, I could have taken over the computing department, but I wasn't that interested! [laughs.]

Abbate:

At Lyons, you mean?

Cooper:

No, at Bliss.

Abbate:

Did you find it difficult balancing your work and family responsibilities?

Cooper:

Well, it could be tricky. I mean you can't just leave them at a loose end. I think I curtailed my hours at the office. I joined them late in the morning and left early in the evening. It was a half-hour journey, and I didn't like to leave the children too long, without supervision.

Abbate:

Did the programmers tend to work long hours?

Cooper:

At LEO, yes. But anywhere else, no.

Abbate:

Oh, really!

Cooper:

I mean, in the early days, there was a lot of night work, that they went in to do testing, because that's when they could have the computer.

Abbate:

So were you ever going in at night?

Cooper:

I never went in at night. [laughs.] I don't know what my parents would have said at the time! I don't think they would have liked it.

Abbate:

Oh, because you were still at home.

Cooper:

I was still at home. I did marry while we were at LEO.

Abbate:

So you weren't going off at two o'clock in the morning to test your programs.

Cooper:

I wouldn't!

Abbate:

But someone else might.

Cooper:

Yes.

Abbate:

And you found out that working at home worked out well.

Cooper:

For actually programming, it worked very well, yes.

I don't know whether you've heard of F International?

Abbate:

Yes. Steve Shirley, yes.

Cooper:

You've seen her?

Abbate:

I spoke with her last spring.

Cooper:

Well, I think Mary told me about her when I was at home, and the information that came back was that if you can work for the firm that you did prior to leaving home, you probably got more money than if you worked for her. But she did very well in her organization— and she had to do everything, of course; she had to organize all her programmers and everything else. I didn't actually contact her, but I think Mary did.

Abbate:

But you knew about her.

Cooper:

I knew about her, yes.

Abbate:

Interesting.

Cooper:

When I worked at Bliss, they used to receive all sorts of computer magazines, which I found—parts were quite interesting. And that's where I heard about the LEO reunions, from one of those! I'd lost touch with them before that.

Abbate:

Were you in any computer societies?

Cooper:

I didn't actually join the British Computer Society, no. I think they were up-and-coming around the time I was giving up.

Abbate:

But you would read the trade magazines sometimes?

Cooper:

Yes. I see one sometimes, if Jeff, my son, happens to have one. I think they have them where he's working, but he doesn't usually look at them! [laughs.]

Abbate:

Did you have any role models or mentors in the computing field?

Cooper:

In the early days, we did. You mean, who did we report to? Who was the expert?

Abbate:

Well, who was helpful, or encouraged you?

Cooper:

Well, it was just the senior staff there. I mean, we were all colleagues together. I suppose we were a little bit in awe of David Caminer, who was the computer programming manager. He was a bit abrupt at times. I seem to remember I reported to him on my Appreciation Course, but at the time I didn't care, so it didn't matter! [laughs.] Probably it came over better!

Abbate:

Right—it made you seem confident.

Cooper:

Yes! I wasn't at all sure that I wanted to join them before that. It was only when they offered me the job that I thought, "Well, why not?" In fact, it was the best move I could have made.

Abbate:

What strikes you most about the way the field of computing has changed since you started?

Cooper:

Well, one's PC is a very complicated program. You're not actually programming. It's a very complicated program—and heaven help those who programmed it!—and it's programming building on programming, of course, to do it. So that when you're working on it, and it doesn't do quite what you want it to do—like, say, Word—you think, "Well, how would the programmer have dealt with that? Oh yes, I'll try this." And it does help!

Abbate:

Interesting.

Cooper:

Whereas you might be completely stuck, otherwise. "Oh, why doesn't it do that? Oh, well, yes, it's the way it's programmed." Like when people say, "Oh, the program is an electronic brain": it's not! It only does what the programmers planned it to do.

Reflections on Women in Computing

Abbate:

Do you have a sense that computing is more open to women than when you started? Or less, or the same?

Cooper:

Computing, you say? Well, more and more people—women and men—are getting a personal computer. Even my younger sister has, who was a nurse; she's got her own now. What was she asking me? "I'm trying to get labels printed, and I can't get it to work!" So I looked it up for her, and Laurie helped me start it. He's more of an engineer-type person, so he can look into the mechanics, and if it's not connected right, he will get it right; whereas I tend to look at it from a user point of view.

Abbate:

And Laurie is your husband.

Cooper:

Yes.

Abbate:

Do you have any advice for young women who might be thinking of going into computing?

Cooper:

Try it. I think women and men are equal, you know, these days—in computing, at any rate. Lots of women now get far more pay than men in computing. Maybe not in other industries—but I think the field's opened. How do you interest them in it? I don't know!

Abbate:

But would you recommend it?

Cooper:

I think so. Yes! If they're interested.

Abbate:

All right. Well, thank you very much!

Cooper:

Okay.

Category:

- Computing and electronics

Contents