Bob Hopgood

Interviewed by

Richard Sharpe

21st May & 4th June 2024

By Zoom

[Part 1]

Welcome to the Archives of Information Technology, where we capture the past and inspire the future. It is Tuesday, 21st May 2024. I'm Richard Sharpe and I've been covering, first of all the computing side of IT, and then the fuller networking as well, since the 1970s. I've also been a programmer, a computer centre manager and a researcher, how about it? Today making his contribution we have a man who is unprecedented, because he did not just give us his background of what he'd done, he gave us a flowchart of his life which he'd done for us, and we will put that up on the site alongside. And let me introduce to you Bob Hopgood, and Bob has a tremendous career behind him, and I'm sure a wonderful one in front of him as well. Bob, you were born in 1935, where were you?

I was born in Baldock in Hertfordshire.

Right. What was the background of your parents?

My father was, family was in the Navy yard at Portsmouth. So my grandfather was basically someone working there and my father therefore went to the local naval school there and did that till he was 15, and then he was going to become an apprentice in the RAF – er, sorry, in the Navy – but it was 1917 so he started off as an apprentice, as a mechanic basically, for engines and as the Navy and the Army both tried to decide which one of the two would have the thing called the Air Force, the Navy sort of thought they'd won, so he started off as a mechanic on aeroplanes. But the naval bit suddenly became the RAF, so by 1917 he was actually working for the RAF doing engines and his first few years he was, he did a certain amount of obviously getting up to spec, and then he went to Egypt and he spent three years there, mending engines, basically. If you wanted to actually learn to fly there wasn't things called parachutes or anything, so you needed to jump out if it was going wrong, and so Egypt was a great place because it has a lot of sand and you can jump out. Unfortunately, it meant that the aeroplane probably got bashed into the ground and so the engine repair business was quite high. I have a load of photographs that my father took of knocked up planes with, you know, either sitting on their tail or on their front and so on. And then when he came back from the Navy eventually he came to

Letchworth and started working for a company called Shelvoke's and Drewry whose major thing they made was dustcarts, which were driven by a couple of paddles that the driver used; one changed the gear and the other one gave you the direction. It was a rather odd way of driving so they had a special test and suchlike and he looked after those until probably the end of the war. Then he carried on doing it after the war. So he had about, his whole career was basically around heavy goods vehicle and their maintenance was roughly how he got going.

[00:04:35]

You were born in 1935, do you remember anything of the war?

Pretty well. I can remember pre-war, which is back. As part of his occupation they decided they had sold so many of these dustcarts to Wales that they would have a maintenance depot down in Wales, and in 1937, I think it was, we actually moved down to Wales from Baldock in Hertfordshire for about a year and a half, I think it was. And during that time the war began to get sort of set up and so things like barrage balloons were coming up and parking at the end of the road, you know, to stop attacks on the docks and suchlike. And at that stage Shelvoke's and Drewry decided to mothball their operation in Cardiff and he was called back to Baldock and I lived there for the whole of the war, basically. Letchworth was fairly busy place in terms of factories and such, which was right next to Baldock and so there was, they had a works road which had sort of Irvin Air Chute on the one end and then a company called Tab, which was the local name for the British Tabulating Company in the middle. They actually, after the war we found out they were making bits for the war effort, so basically Colossus and things like that were being manufactured at the Tab. And then farther on was Shelvoke's and Drewry, and at the far end was a company called K&L, which was taking basically any piece of iron they could find anywhere that wasn't necessary and turning it into steel. And that got bombed quite a bit, the Germans tried to bomb it, but they seemed to fail totally, but quite a lot of bombs landed in and around Baldock. Things you did in the evenings or the weekends was going and look at the bomb holes. So two years doing that.

Did you enjoy school?

I can remember all of that.

Did you enjoy school?

Yeah, I went to the junior school at Baldock. We had a set of ladies who taught us through the early years and half of them we spent out in sort of shelters out in the back underground singing songs, as far as I can remember. So it wasn't until 1945 that the thing got back to normal. But they gave us a reasonably good education so there was nothing wrong with them. Just after the war I passed the exam to get me into Hitchin Grammar School, which was the local grammar school of merit, or so they thought.

Were you athletic?

Say again?

Were you athletic?

Pretty well. Yeah, I played most games, so at Hitchin Grammar I played football in the first term, hockey in the second term and cricket and tennis in the summer term there. I also played table tennis probably every night of the week, so yeah, I did quite a lot of, both at school and at university.

You…

So I was going to say, the two main ones was table tennis and hockey, they were the ones I was better at.

You focussed on science did you?

At school, at the grammar school, it was basically doing maths, physics and chemistry, the standard set that you were getting in those days, and probably I was better at the maths than the physics and the chemistry. So I think I landed up with

five A levels, which was maths, physics and chemistry, and then advanced maths and further maths, so I got three A levels in maths and the other two, yeah.

[00:09:42]

You then for three years, did you not, you went to Kayser Bondor?

Yeah, that was a sort of, that was my first excursion towards sort of knowing about computers. When I was 15 Kayser Bondor was a large factory in Baldock, so Baldock High Street is about 400 metres long and all along there is the factory for Kayser Bondor who were, during the war were making parachutes, and then they went back to making ladies' lingerie as soon as the war ended. And by the time I got to 15 they were pretty back to normal and starting automating their basic buying and selling part of the business. So it was a sort of 400-metre long factory and at one end there was a warehouse which kept the goods that were being made, and a finance department, which basically they took orders from shops around England and produced them, sent them an invoice for the money, and they paid it, hopefully that was the way. And the way it was organised was sort of computerised in the sense of they had about six ladies whose job was to take all the orders in and to punch them onto 026 IBM, with 026 IBM card punches, and eventually they would get a deck of cards which (a) said which of the many companies had ordered a thing, which shop had ordered it and what the things were that it wanted. This produced a sort of small deck of cards. And so there were four to six ladies, I can't remember, it was at least four on these 026 card punches, which were basically getting all of the orders in, and at the end was some variety of card punch machines, including, I think it was an IBM 062, if I remember, which could add, multiply and subtract, and effectively you could do a program, you could put a plugboard together which produced a program which either could add up and say these are the things that need to come out of the store. And it had another program which basically said now the store we are low down in these products and sent off a message to the factory saying this lot needs to be… So that was the input side and then on the output side the, hopefully, the cheques arrived and there was an equal number of ladies sitting on Friden Flexowriters on the other side of the building and they punched in the things that came back from the customer and… Yeah, so there was an office at the end which contained a lady who had a very

fancy Monroe calculator which could take Flexowriter tapings in and add all of the stuff up and eventually, well, cutting a long story short, the only problem was, often the money that got sent or the cheque didn't actually work out exactly as what it should be. At first I was allowed to take discount if the order was, the cheque came in within a certain number of days. So that was the input. Every now and again you used to get these things, they'd cancelled one thing, they'd sent one order back, and so there was a complicated arrangement and they sent a cheque and the cheque bear no resemblance to what we thought the cheque should be. And being good at maths, the guy, I knew the guy who was the finance officer running this whole establishment, and he'd say, will you come in on Saturday mornings and you can puzzle out how on earth this person came to this amount. And so that was my role initially. And then later on I used to go there in the holidays as well. And then if anyone was on holiday I would take over their bit of the finance chain. So I got to know the ladies on both the 026s and the Flexowriters when we were trying to work out what on earth the person had done to send this money in return for what he should have been putting in. And eventually after the Saturday morning we got it all sorted out and then we started another cycle of this. So that's how I got to go there. Later on I heard Kayser Bondor had removed the card punches and got a LEO machine in. So that was my first venture into anything resembling computing.

[00:15:28]

And then National Service from '54 to '56 in the RAF?

Yes, that's right. I had two years… I got an offer from Cambridge for a place there, but I didn't get an exhibition or a scholarship or anything and, you know, the university one, so they made you do the two years' National Service before you went up to university. And I got trained on SWAB 8s which were sort of from here to Singapore, the transmitter could transmit six, I think, six teleprinters' input all on the one channel. So at one end you had six teletypes putting information in and say, Singapore or Australia was where the… and that was basically the long distance traffic. So we all got trained down in Berkshire and further on to Compton Bassett – that's right, I was trying to think where it was – and we got trained and then I think

two people went to Australia, two people went to Washington, two people went to Singapore and the other two or three of us went to Sheffield.

[RS laughs]

We landed up in Sheffield and basically, we were supporting brand new automatic mobile radar systems with some air-to-ground requirements. So if the person was coming into ground or whatever, you could get involved with them. They were brand new, so we didn't actually do very much other than sit there and just make sure everything worked. I did that for about half the time in the RAF after I was trained, and then finally got to Watton in Norfolk and there they did have a SWAB 8 transmitter, so I spent the last few months actually working on the system that we'd actually been trained for, but the prior bit wasn't. Watton was quite good, it had Canberras and Vulcans and it also had some old Lancasters that were falling to bits. I notice one is still running today, there was these Lancasters around as well. So yeah, it was an interesting time. Didn't really get to know any computing or any involvement in computers during that period.

[00:18:32]

But then you went to Cambridge?

Say again?

Yeah. You went to Cambridge in 1956 to 1959 and studied mathematics there, is that right?

Yes. I studied maths for three years. And it was, maths in those days was, well, basic mathematics, learning how to do matrices and do this, that and the other. But there was a certain amount of emphasis on computers, so we got taught about, Hartree taught us things that we ought to know about if we were trying to solve differential equations, etc. And so there was quite a lot of theoretical stuff in the maths course which were relevant to computing. And then the actual computing bit was, the maths degree was just the thing that you did in the afternoon, basically once or twice, plus a

certain amount of talking about computers. So Wilkes and Miller were the two computing people who gave lectures to the maths set. Now I think there was one or two practicals in the afternoon, mostly on broken Brunsvigas, so if you tried to add, subtract three times something, you had to, as you couldn't use the '3' button you had to do '4' or something like that, and then add one. So they weren't, compared with the Marchants I'd been using at, you know, electric Marchant at the Kayser Bondor, they were pretty awful. I guess people who were doing computer science, it didn't really exist as a degree, but if there were anyone attached to the EDSAC lot and that, then they presumably got decent machines, but poor simple undergraduates didn't get very much. So I didn't really, I learnt a lot about things, how to run, you know, what numerical analysis was needed, but that was practically all I got on the computing side. It was interfering with hockey as well, because I played hockey in the afternoons and practicals were in the afternoon, so I missed a few.

[00:21:20]

1959, you went to AERE Harwell. 1959 is the year of COBOL and also the year of the Planar technique for making semiconductors, which is rather important to us all. You went to Harwell, Atomic Energy Research Establishment, is that right?

Yes, yeah. I mean the Atomic Energy Authority had effectively a research establishment at Harwell that was looking at peaceful uses of, ie making piles to generate electricity, that was the goal there. And at Aldermaston down the road about 20 miles was where the weapons side was being done. And there was an offshoot at Winfrith which was near, basically doing design for a genuine atomic pile for going in the grid and that. And in the north there was a Risley one, a site which did the actual work of getting piles running and getting them out into industry and running up. So that's how the thing was generally laid out. So Harwell was the sort of peaceful research bit, long-distance research. Apart from it had one of those things was thermonuclear fusion, so most of the fusion work when I was there was at Harwell and it was spun off into another establishment, sort of in the middle of when I was at Harwell. So basically, at Harwell my job was to – I was in the computing section, recruited for that, and we had a Mercury and within the Atomic Energy Authority at the start Aldermaston always had the best of what was available, so that they had an

IBM 704, which Harwell had access to, so I could program both the Mercury and the 704. And the 704 turned into a 709 and the 709 turned into a 7090 and eventually into a 7030 Stretch, so that was the sort of… so remotely, mainly by card input and output via taxis was how we got jobs run at Aldermaston. And then on the local system we used Mercury, and so Mercury was the main source of my computing experience at that stage.

[00:24:24]

That was a Ferranti machine, was it not?

Yeah, it was a small, well, called small, but it was quite, it was amazingly quite fast compared with its competitors at that time. And they sold well abroad as well as in England, so the Canadian Atomic Energy also used Mercury computers, and I think Shell in London had a Mercury, so there was quite a lot of Mercury computers around, they were relatively fast. Quite small, 1,000 words of memory, if I remember right. Compared with today, it wasn't a great deal. And basically the computing group, the programmers mostly used Mercury order codes for coding problems and people who wanted answers to, like, can you do this integral and tell me what the values are between and print them out for me. They'd come out and it was sort of like a shop, and the next person who wasn't doing anything would do their calculation for them. So lots of little programs on Mercury for about the first six to nine months, and then I got involved in some larger projects. There was initially one where they had a program from Chalk River. Basically, the research group at Harwell just got the remains of any piles that were being used professionally for testing stuff, so the old GLEEP and things like that, and they did the same with the fuel rods. So they got, well, when someone had finished with a fuel rod they picked it up and then they had to work out what on earth was in it, if they were going to do an experiment with these half-finished fuel rods. And they had a program which came from Chalk River which effectively did this, and it ran ever so slowly and so as a result it was using a lot of Mercury time, and they sort of asked me if I could actually make it run faster. And I roughly knew what the program was, and the only constraint was that I wasn't able to do anything in terms of modification other than taking the bit of tape that came from, this binary tape that we got from Chalk River, and to punch some holes in the end

which would do it, so it was quite a task. So I'd had to disassemble this whole program and work out that it was using Runge-Kutta for doing something and I changed it from basically being involved on [incomp] to e minus, e to the x, and as a result the program ran faster and now I had to punch it onto this bit of tape because numbers in the actual program were confidential and we weren't allowed to change them or see them, so that took me about six months, I think. And then we, I guess the next year or two, the last period before I went to Aldermaston was with the Quantum Chemistry group, and then there was a large program, Fortran program that ran on a 7090 or a 709 and I basically changed that so that it would work on larger molecules than the ones that it was being used on. There was a big thing called the Quantum Chemistry Program Exchange, which basically gave young people working in that general research area, help, and MIDIAT was the one I got involved with and worked on that for probably a year or so, mostly looking at manganese oxide for some reason.

[00:29:00]

And then three years at the Atlas lab?

Yeah, well, yeah I had, well, after the first few years I then went to Aldermaston for a year, which was mainly, they'd got the IBM Stretch and Harwell was working towards getting their machine, which was going to be an Atlas, or so Harwell thought. So the trouble with the Stretch was it arrived and it was assuming effectively that it was mainly going to get programmed in machine code, and Aldermaston had made the decision that as they kept getting new machines and different clients, they were going to concentrate all of their future programming on Fortran. And a decent Fortran compiler came from IBM, but it was so slow at compilation that it was running about a quarter or a fifth slower than the 79, which was hardly progress. So they did a quick dirty Fortran compiler called S1, which compiled very quickly, but didn't produce great code, but it allowed them to then move a load of programs from the 709 onto Stretch and then they decided they were then going to make another compiler, S2, which would be an all-singing, all-dancing optimised compiler for Stretch. And they didn't have all the staff they needed so I got seconded there for what was supposed to be four months and eventually turned into eight months, helping to write the S2 Fortran compiler, which is how I got into the kind of area of computing that I

eventually landed up in, and it was mainly through the period of Aldermaston that that happened. And…

Then Atlas.

Say again?

Yeah. And ALGOL.

So when I got back from there, the decision had been made about a year or two earlier that an Atlas would be installed on the Chilton – Chilton is the name usually given to the whole of the Harwell site. So it was going to arrive and be on the Chilton site and the question was, there was Harwell and the Rutherford Lab next door, and the two of them wanted more computing power and it wasn't clear who was the one who ought to have it and at the same time the universities were really short of computing power and everyone was starting to use ALGOL and teach computing properly and all that, and as a result after about three years of negotiation it was eventually decided that the Atlas Lab would be set up as a separate entity and provide computing to Harwell, Rutherford and the universities. So that's how the Atlas Lab got positioned where it was, halfway between the Harwell part of the Chilton site and the Rutherford part. Rutherford was doing high energy physics and Harwell was doing what I mentioned earlier. So that was it. Harwell had made, I guess, two years before I went to Aldermaston, a decision that they would write a Fortran compiler, because the university at Manchester that was helping Ferranti in building the Atlases wasn't interested in Fortran. It had the Atlas Autocode, the Mercury Autocode and things like that and they were making compilers for these using the Brooker-Morris compiler-compiler system they had, which was a program for writing compilers in. And they'd decided they were going to do an ALGOL one and the only trouble was, ALGOL was a funny language in the sense of it was very elegant and ideal for teaching purposes, but it had a number of unusual features, like it had no input/output commands. So as a result every manufacturer had their own input/output commands, which wasn't a great asset. And the other thing was, it basically had words like 'procedure', 'begin', 'end' and 'goto', etc, which were reserved words. And you had to sort of differentiate the reserved words. So each of the manufacturers produced a

different version of ALGOL, so KDF 9 used eight-hole paper tape and you could actually do an underline before the character, so if you wanted to do 'goto', you did underlined G, underlined O, etc. I think some of the others you could actually backspace and then do all the underlines at the end. Some just gave up and said, well, the word 'goto' is going to be reserved and you can't do anything about it, sort of. So there were loads of dialects and each was separate. So Bart Fossey and I went up to Manchester and said we're going to have to provide ALGOL versions for all of these machines that are in the universities, and there was Stantec Zebras, Elliott 803s, KDF 9s, etc, all with different versions of ALGOL. And so we went up to see the ALGOL compiler that was being produced by the Brooker-Morris system and talked to them and said well, can we sort of put a front onto your thing and get into- they had a two-part series whereby they did the lexical analysis for the ALGOL and then did they ALGOL actual compilation of the code itself, and we could sort of slip in after they'd done their bit for the seven-hole paper tape that they were using. And basically they said over our dead body are we going to let you anywhere near our compiler. For a couple of reasons: one, we've had experience on Mercury where universities messed about with things and messed them all up, was one of their statements. And the other one was the last thing we want is to allow other people's company's, programs to run on the Atlas otherwise we won't get any Atlases sold. So they had those two problems. And we talked to them and we went off again a couple of times and eventually they said, oh okay, we believe you understand what you're doing, we'll let you in. So they wrote the basic ALGOL compiler for Atlas and Alex Bell and I basically produced a version that would accept any dialect, but which was mainly KDF 9 and Elliott 803. And then injected into the second path of the ALGOL compiler that had been made at Manchester, and then we put an outer block round the ALGOL program and in the outer block we had all the IO procedures for whichever IO system was needed. And that worked pretty well.

[00:37:46]

Do you have a favourite programming language?

At the moment?

Yeah. One that you favour in your career?

I'm not sure I deserve to answer that, it might incriminate me.

[laughs]

If you ask me to write a program tomorrow, then I'll write it in Fortran.

Okay, that's fine. It's a very elegant language, isn't it, Fortran?

Well, Fortran was a language designed by people who wanted to run large programs, so I mean the two things about Fortran which are brilliant are, one, you can sub-compile. So the standard thing is, you write a load of subroutines, the subroutines get compiled, turned into binary which is relocatable, and so effectively once you've got a routine that, you know, works exactly as you want it to, then you turn it into binary and the card deck you have goes down a lot in size and eventually you can give those subroutines to other people and so on.

It was the third programming language I learnt after, what, ICL PLAN. You probably remember PLAN?

Yes, yes.

And also the Job Control language for the 1900, and then Fortran, and then COBOL.

I used PLASYD quite a lot as well, which was the sort of high-level assembly language on the 1900s, which was a language. Because most of the early languages are sort of not available, apart from Fortran. There are one or two quite decent Fortran compilers still around. At the moment, as I still have an interest in trying to see the history of, you come across Fortran programs that you can… and if you've got a compiler you can actually work out what they were doing and so on. So that's my spare-time hobby.

You spent a year in Pittsburgh, why?

Say again?

[00:40:20]

You spent a year in Pittsburgh.

Yeah. By the time it got to ’66 – can I back up a bit? We decided we’d send people to IFIP in 1965, that was being held in New York. And so Bart Fossey, Bob Churchouse and I did an extended stay and we basically visited everywhere in the US which was of any interest. So we did MIT at Boston, we went to CDC in Minneapolis, we went to Stanford and found out about Niklaus Wirth and he was going to develop Euler and Pascal, and McCarthy was there doing LISP. And we also visited Carnegie Mellon at the time, which had people like Herb Simon and that doing Artificial Intelligence. Our first was into compilers, and so it was Bob Floyd and… So it was a nice place to have, we enjoyed the visit there, and it came out and said, well, can we have an exchange. If you want someone to come over and stay at CMU for a year, or Carnegie Tech as it was called at that time, well, we’ll take one of your people. So I got the designated job, I was the lucky one to go. So I had a sabbatical there for a year, which was nice. And yeah, that’s where I spent a year. So we did the 1966/1967, September to September, basically, in Carnegie Tech, which during while we were there turned into Carnegie Mellon University. It was good, but there was a lot of impressive people around. There was Herb Simon, Al Perlis, Gordon Bell.

[incomp and no sound briefly]

[00:42:53]

So I went to a lot of courses, which was nice, and I had to do something while I was there and they were building a great big CABAL compiler-compiler system and it was soon clear that it would take at least two years for them to finish it, so instead I changed it and decided I would rewrite the compiler-compiler for the machine they had at the time, which was the Bendix G-21, and then the idea was it could get ported to the 360 series later if they ever got them. So I did that.

[Question? Sound distorted]

[00:43:44]

Sorry? It was a sort of an interesting exercise, because all I had was the code for the Atlas, the compiler-compiler, I'd got a listing with me and the machine I was going to run it on was a Bendix G-21 which bore no resemblance to Atlas whatsoever. And I came out with the idea, well if I actually input all of the Atlas code and run it as though it was a Bendix G-21 code, then every instruction would come up with an error, basically. So I thought well, it doesn't help a lot. But you could write an error routine into a writer program to intercept the error and do something, so you could actually… So I fabricated a program which had the same structure as an Atlas program and effectively, every command that the Bendix G-21 rejected, I then turned it into what an Atlas program would have done with it. That actually worked quite well, it was a bit slow but it meant that I could actually get the whole of the compiler-compiler running on a Bendix G-21 and could actually do some work as though I was writing a compiler for that. So that took me about the first few months and they had some Philco, big Philco displays interactive displays which they were using for things like interacting with chess programs and other programs. Herb Simon was into those, draughts as well, I think he was. So they were quite interesting things that they were doing. And eventually I got round to writing sort of graphical output routines for a language called Formula ALGOL that they were developing. In Formula ALGOL, if you write A=B+C, then it can either mean that you add B to C and deposit it in A, or it can be A is the formula B+C, so you basically had an algebraic system or a normal numeric system, whichever way you wanted to do it. So you could do things like integrate something and, you know, you could integrate x cubed to get x to the fourth over 4, and so on. So it was a nice system. The only trouble was, they couldn't print out any of the results, because printing out mathematics isn't that simple, so I did some work whereby the Philco displays could actually display the results of the Formula ALGOL system, and that took me quite a while too. And I did that using the compiler-compiler, which was nice. So it was a good year away.

[00:47:09]

And then back to Atlas?

Yeah, by the time I got back to Atlas, (a) the Atlas was running fine, bedded down, ALGOL user came up to me on the day I got back and just asked a query about something, and it was just as though he hadn't seen me for 13 months and it was the next day, kind of thing. But basically, yeah, the ALGOL system was running fine. The number of dialects we were now running was over ten, I think, so it gradually got bigger and bigger. And we had it in French and Danish versions and programs, all manner of things, but mostly Alex Bell had been doing that while I was away for a year. But in the meantime, we'd just got a SC4020 microfilm recorder arrive, that was the big new thing, if you like, at the Atlas Lab. And SC4020 was basically a machine with a graphics machine which could draw graphs onto either 16mm film, 35mm film, or some hard copy output, and therefore you got the opportunity and the ability to produce computer animation. And as that was something that had always interested me about the possibility, I was, decided that what I'd do was write a system called GROATS, which was Graphic Output on Atlas used in the SC4020, and that would allow people on the ALGOL side to be able to write programs to generate graphs or films. And the same- and Paul Nelson, who'd got the, was actually getting the thing up and running, the 4020, did the same thing for Fortran. So basically we then got into about two years where we spent telling the world how to make computer animation using an SC4020. Both Paul and I went round talking to universities trying to get them interested and personally I produced some films for the A Level physics course, the new Nuffield A level physics course on Change and Chance, which was very integral part of that particular programme. So the new A level course was partly dependent on the computer animation that we did. The OU was doing a new maths degree course which they wanted lots of animation to be available to them each week, and we spent some time actually seeing if that was possible on the 4020 and decided it was, and a Brunel undergraduate and Tony Pritchett, between them they did one week of computer animation for the maths course every week. So it was a demanding time, a lot of animation came out of that. So that's how I mainly got interested in the next few years, doing computer animation for various projects.

[00:50:53]

Okay. I've made an executive decision. We're going to actually, Bob, if you agree, do this interview in two parts. We're going to carry on now for about half an hour to 45 minutes, if that's okay?

Yeah.

And then we're going to come back to it on another day – you choose the day – and do the second part of your career, because we're getting a lot of rich material out of this and I don't want to press you too fast. Okay? Is that alright?

Yeah, yeah, yeah. I had a feeling it was going to run over, because, well (a) I talk too much, which doesn't help, but…

No, you don't, no you don't. 1972 to '75, what were you doing then? You were doing that computer graphics were you?

From… I'm just trying to think what I did. So basically '68 was the year – '66/67 was the year I was away – and we carried on for the next two or three years doing computer graphics mainly. We had a PDP15 front end to the SC4020 and the two together gave us the ability to (a) generate quite a bit on that. And around about, must have been about '71 I would guess, Bob Churchouse left to become a professor at Cardiff, and he was running the software group in Atlas still from the beginning, and so I took over the role of doing the systems side of that. So I was then turned into basically a group leader for what was called basic software. And Barbara Stokoe was the one who dealt with the application side, so the two of us basically took over what Bob Churchouse had been running. And so the last few years of the Atlas system was me in charge of basic systems, which was (a) the graphics stuff that I'd been doing, but also the decision had been made for a replacement for the Atlas computer. We'd started around, I don't know, '66, '67 looking for a replacement for Atlas which was getting a bit old in the tooth. And if I go through and tell you the history of that we'll be here for about another week, so this is the shortened version. We wanted to get the largest and best offering that we could get from someone to replace Atlas. So it would be nice to get something ten times larger. And I guess the best one we could

find was the 7600 which was announced just before I returned from Carnegie. I drove up to Minneapolis and back to get a pre-launch version of what the 7600 was going to like.

From Control Data Corporation?

[00:54:24]

Yeah, yeah. So I went up to Minneapolis, drove all the way there in a day, which was about 800 miles, and came back a bit more leisurely. That was just before I came back. It was clearly probably a possibility. And the Department of Industry was determined that it ought to be at least partly British, if not all British. And ICL said they, well, ICL were in the middle of going from being just them and gradually getting other companies, so they swallowed up eventually the English Electric side and the Elliott side, and so ICL had got to the stage where they'd got a lot of machines, all of disparate types. Some were IBM compatible, some were, you know… Net result is that they needed a new range of systems, but at the same time that would take a few years to actually get through the system and in the meantime they extended the 1900 series upwards as far as they could go. So the offer we eventually got was a dual processor 1908A, which the DTI was happy with. We weren't that keen, but it was a possibility. It was a 6-bit architecture, rather than 8-bit, which wasn't great by then, but it was clear the new range was going to take longer than what was necessary, so a dual processor 1908 would have been significant amount of computing power. And they had grandiose plans of having one operating system running the two processors, which never actually worked or got going. So a 6A was ordered, and so we had a 1906A on order with the potential of changing it into a dual processor 1908A and DTI was happy with that. And then ICL decided that the new range that was being developed with Manchester might come along a bit earlier, and so they cancelled the 1908A. So Atlas was left with a new machine which was a 1906A which wasn't a great deal faster than the Atlas itself and no likelihood of a 1908A. So we then looked at buying a Star 100 from CDC or 7600 to backend the 1906A. And in the end, the money wasn't there and the decision by SRC was to close the Atlas Lab and effectively merge it with the Rutherford Lab next door, who'd managed to persuade everyone that they needed a lot more computing and got a

360/195, which was the fastest thing around, you know, faster than Stretch even. And Rutherford offered 20% of that machine, so the next few years I was left with getting a 6A up and running on George 3, George 4, whatever, getting all of the machines that were currently in there either sending work to the 195 or to the 6A. And we started getting basically the remote job entry sorted out. So that was probably the three years up until the merger with Rutherford. And then we, basically everything changed completely as a result of that.

[00:58:53]

Why did they merge?

Well, they were basically short of money, I think is probably the answer. It normally is with Science Research Council. When things get bleak, and they often do, either the government cuts the research budget, or the Swiss franc and the pound change in value. And that has a big effect because the amount of money that was paid each year to CERN for use of the CERN facilities was quite a large amount of money, so it could vary by hundreds of thousands of pounds in a currency change, so there was always the problem that either the Rutherford Lab had money or they didn't have money and so on. And eventually, I mean the universities had by that period got some really good computing equipment. The Flowers Report had done a good job, so the top end of the university sector were getting, they'd got KDF 9s, lower down they'd got 803s and new systems were coming in and the actual importance of the Atlas, it had gone from being one where probably most, you know, a significant part of the computing in the academic centre was the Atlas Lab, and it had got by that period to being almost irrelevant because the Flowers Report had turned regional centres [incomp], so the London, the Manchester and the Edinburgh systems were all now significantly as good as, if not better, than what the Atlas Lab had. So the Atlas Lab's reason for being was either it was going to have to be a national facility, and DTI at one stage, just before the merger, had proposed that instead of merging that they would turn the Atlas Lab into a national facility which would run the largest and best of any UK products. And they would actually take the CAD Centre and move it to the Atlas Centre and sort of bolster up as being the sort of flagship. This was all, even when the two merged, this was still sort of in theory what was going to happen,

and as things went on it was gradually forgotten, mainly because there was a shortage of money and also I think the Rutherford, it had got a significant amount of computing power with the 195 and Harwell did the same. So Harwell got their own computer and Rutherford had got their own computer, the university sector was- and so the decision had already been made that instead of any computer person in a university could use Atlas, it was now just for grant holders, so it had gone from one stream to the other and I would guess it was the right thing to do at the time, given the finances, but wasn't very happy for the people who worked for Atlas and quite a few of them left. Some were moved to- science support work was going to be concentrated at Daresbury, so some people actually moved up to Daresbury, some left and some rather disgruntled people became merged with the Rutherford Lab, but it took a while before we got over the changeover, I guess.

[01:03:03]

Then you changed in 1975 to 1979, what were you doing then?

'75 to '79? Well, when the changeover came, effectively Rutherford had their own computing section called Computers and Automation, or C&A for short. That was one and effectively the part of Harwell, part of the Atlas Lab that was dealing with the running of Atlas and the 6A became part of that division. And at the same time, SRC was having a lot of remarks from the engineers saying you're not giving us enough work and there was a Rosenbrock Report that came out which basically said, (a) engineers aren't interested in old-fashioned batch computers and what we want to do is interactive design and what we need is small systems in our departments, and please could you provide these. And so SRC actually had so much pressure from the engineers that the name was changed from SRC to SERC, Science and Engineering Research Council. An engineering board which was running outside of it came up with the Rosenbrock Report, which basically says what we need is small multiuser systems situated in university engineering departments so that they can get on with what they ought to be doing and set up this interactive computing facility will you please. And that was the sort of remit for the software side of the Atlas Lab that moved across. So there was a separate Atlas computing division which Geoff Manning, the deputy director ran, and basically all of the basic software group and the

applications software group that Barbara Stokoe and I were running, became part of ICF. So one day we were looking after FR 80s and microfilm recorder that we just replaced the SC 4020 with and turned into, well, we're now into interactive computing for engineers and that's what happened after the break, you know, the changeover. We ran the ICF I think from 1975 to 1990, so it was quite a long programme, and it basically says, upgrade any of the engineering facilities that we've got at the moment till they work better, then gradually add multiuser systems in all the main engineering departments and make sure that they're capable of running six to eight people at a time doing interactive graphical programming. I mean that was the sort of ICF remit, and to get the applications side of it so that everyone else was going in the same direction. So that was the sort of remit for the Interactive Computing Facility. And I guess for about the next 15 years, that was a fairly central thing, although the initial three years or so were quite demanding because (a) there was basically two large PDP 10 systems which were at Edinburgh and UMIST and those two were the main source of interactive computing in the engineering side of things. There was an Interdata machine somewhere, I can't remember where, I think there was a Prime at Nottingham, but that was it in terms. So the way the ICF got set up was initially to upgrade the two DEC10s, having done that, upgrade any of the facilities already in the departments that were working, and once we'd done that then to look at what if anything was sensible in terms of a multiuser system capable of doing real work with graphical terminals, etc, for up to six to eight users was the kind of demand. And we went out to tender and there was a lot of people who thought they had such things. Even ICL made a bid for one of them. Some were soon dismissed, but out of it came Prime as being one of them, the main companies that could deliver. Clearly DEC could deliver with their PD11s. We'd got a GEC machine that was front-ending the 195 with the 1906A and that looked as though it was a possibility too. And DTI as always was pushing hard that it should be a GEC British machine and not the US Prime one. And so after a lot- we had a real problem with benchmarking these things because there was no benchmarking software. With DEC, I could walk into DEC in Maynard and say, I want to buy a DEC10 that will satisfy 60 users with this profile and tell me, and they had a, basically you could simulate the whole thing, and it was great. Whereas there was nothing. So the only way you could actually test any of these systems was actually to get six people to actually sit at terminals and run a script like that and see how long it took. And the idea was that if

it got below half an hour, you know, we had some heavy computing jobs as well as ones demanding on a graphical output side and so on, and we all sat round and went round the countryside putting benchmarks on various systems. And out of it came, I guess the GECs just got through and the Prime 400s were easily the best of the offering. And DEC, which could have been one of the suppliers, declined to actually provide us with a benchmark with the Unix system and insisted that the only one they would offer was their, with their native software rather than Unix. It was a pity really, because it was clearly, later on, would have been a really viable option, but the company declined to do it and so that was it. So we bought Primes and GECs. I guess there was maybe 20 systems that eventually went into universities. So it was a big programme, the ICF programme, and it concentrated on putting those machines in and getting them working. If you've got that many machines trying to access a 6A and a 195 and remote job entries here, there and everywhere as well, we actually had a network. And so we started off with a load of independent little bits of a network which eventually got all put together to become the academic network for the UK. So that was, one of the main inputs, I think, to that was the ICF which had all these scattered outside the computer centre systems that needed to be integrated into the whole academic field. So it was part of that and did a good job on it as well. So eventually…

[01:11:53]

This network was the precursor of JANET, was it?

Yeah. Well, it started off with various names. I mean and we started off with all the Primes being able to talk to each other so that we could actually develop a tool that finds… basically all the keeping the operating system up to date, etc, was done centrally and we sent it out, and the same with the GECs. I mean we had control of both those. And the GECs were the only ones that ran the JNT preferred set of, Joint Network Teams preferred things to go with the ISO standards, which was a perfectly valid way forward, and so the GECs all ran the JNT Colour Book set, if you like, whereas the Primes, the DECS and everything else, for everyone else we gradually moved on until everyone was running as JANET network set of protocols, which, as you say, became SRCnet, so JANET or SRCnet or whatever you like to call it. And,

you know, the communications side of the universities at that time was quite good. I mean I think it was as well up as anyone. The only thing was, they didn't really get involved with Unix and TCP/IP, etc, which not regarded at that time as being the way forward.

Which is ironic, because in 1975 when this starts, was the first connection across the Atlantic, London and Stanford, with TCP/IP.

Yeah, yeah. Rutherford were actually part of that. I can't remember exactly which bit we did, but at some stage Peter Kirstein's machine got involved and it was funded by the DCS programme. So we had that and I think one of the satellite bits was also with some SRC physicists involved in it, I can't remember which bit of the tracking round from US to Norway and back, and back onto the… But yeah, Peter Kirstein was the main one who was regarded a bit of a renegade who wasn't doing what everyone else thought he should be doing, but that was Peter.

[01:14:40]

Why do you think the ISO standards, OSI standards fell away?

Mainly, I think, because of US. US were not going in that direction and therefore, you know, eventually the US will win because of just sheer size and the money going into ARPA and suchlike and ARPA were pushing that. And it happens, not just there, it happens in a lot of other areas, you know, if you get a bandwagon and the US are going one way, then it's going to be the way that everyone goes eventually, or it used to be, yeah. So Unix caught on and everyone, eventually CERN had Unix access, so did Rutherford and everyone else and eventually the move came. And it was soon after this period.

And with Unix comes C, and to me, C as a programming language is, well, it's a dog's breakfast, isn't it?

Say again?

C, the programming language, C…

Oh, C.

It's inelegant, it's a dog's breakfast.

Yes. I mean the C language came around with this Unix lot and it's a bit sort of assembly-like in the sense of you can sort of work out what the code will be that the C program generates, and so it's got a lot of, it had a lot of advantages because of that. Plus the fact that it ran on Unix and everything. So we got heavily involved in C for a while. But yeah, it gradually got taken over by- it's like with everything, if you get a decent language then someone else will produce a better version of it and change it and I think there was… the software engineers were saying it ought to be something like Ada which was much more well-defined. Other people who liked tacking around much preferring C, and so on. You got a great number of languages coming and going. Pascal. Euler. You name it. Seed…

Have you ever used APL?

Not in earnest, I think I wrote one program once. That was the sort of ultimate in preciseness and smallness, yeah. Yes, it's a nice elegant language, but it's like Cleo 1, it never really managed to get to the stage where everyone was doing it. I think to be fair, with C, I think it was, I remember standing in a queue for one of the university graphics conferences I went to, and I was trying to register and there was a couple of guys standing in front of me from Bell Labs who said, you know, they'd started up this idea of having a language, an operating system, just for fun, just to show that it's possible to do better than GE were doing on their large machines. And it was Ritchie and his colleague and they were saying, then it caught on and we got lumbered with it, sort of. I think they started off by doing it just as a demonstration that you could make quite a sort of very low effort operating system that would work and they were very successful.

I think that's a good place to pause, Bob, if that's okay.

Then we go to Unix soon, yes.

Yes. I think we'll take that as the end of part one of your contribution to the Archives. Thank you very much for your time, Bob. Is there anything in part one that you think is missing?

I think we've covered most of the areas. I'm trying to think of any others that we haven't. There's probably… we've not covered really that there was a lot of applications work going on as well as the… so I've concentrated on the bit that I've got involved in, but I think Atlas and that whole area up to the Engineering Board, there were some large software libraries and stuff that were being developed on Atlas which are still around today and still being worked on, so yeah, the applications side, which wasn't one I was involved in, we probably have undersold what was going on. And there was a lot, there was a lot of application areas that did significant work.

Okay.

I think that's probably the main thing that we- but then that was mainly because I wasn't involved in them. Barbara Stokoe's group were involved in that.

I want to focus on what you were doing. We will end part one here.

[end of part one]

[Part 2]

[01:21:05]

Welcome back to the Archives of Information Technology where we capture the past and inspire the future. It's Tuesday, 4th July, 2024. I'm Richard Sharpe and I've been covering and researching IT, first on the computing side and then later in the full IT area, since the early 1970s. Indeed the first computer I saw was an ICT 1902A used in AMI Electronics and Research Division. I say welcome back, because this is part two of the contribution, a very rich contribution so far, and I'm sure it'll be even richer in the second part. It is the second part of the contribution to the Archives by Bob Hopgood. The story so far, this is what we want. Bob was born in 1935 when Alan Turing was mulling over ideas which came out as the paper on computable numbers, and also on the other side of the pond, a man called Claude Shannon was mulling over ideas and later on in the next year, 1936, they both published very similar works, Shannon's being on the use of Boolean algebra in the logic of switching – switching for telephones. Bob went to school in Cardiff, then Hitchin Grammar. His first real piece of, I would call it data processing really, was because he was good at maths and they knew it. He went to a local company and helped them with their accounts, to reconcile their accounts, and that was really a data processing job, and he did that part-time from school. He was then sucked into the RAF for his National Service for two years, and then from '56 onwards he did mathematics at Cambridge. In '59 he joined the AERE, Atomic Energy Research Establishment, at Harwell, and then '63 the Atomic Weapons Research Establishment in Aldermaston. In '63 also he joined Atlas Lab, the Atlas computing lab, great big machine of its time, a supercomputer of its time, built by Ferranti, and with paging, for instance, and he learnt ALGOL there and also he writes programs in Fortran and others. He spent a year abroad in Pittsburgh in '66/67, travelled around the USA, and when he came back he went back into Atlas and did work on graphics, and also from '72 to '75 basic software.

There's something that I want to pick up right away, we're really starting the story in 1979, but we've skipped one big part of your story, Bob, in the first part of our talk,

and that is when you came back from Pittsburgh in '68 – '67/68 – you started teaching at Brunel. What was that link, how did you do that?

Well, it was basically when I was in the USA with the year off, I went to one of the…

You've frozen.

[01:24:30]

…bump into but Mike Pitteway. Mike Pitteway was currently computing, running the computing side at Nottingham and had just decided he was going to take up a new post at Brunel and was going to set up a computer science degree there, and I just happened to bump into him and he said, oh, do you think you could come to Brunel for a year just one day a week, do a course for me while we get started. And I did a compiling techniques course with him and next year he said, well, we're not still ready and so I did another year, and 30-odd years later I was still doing either one or two courses per year at Brunel University. He's quite persuasive, Mike Pitteway was.

Do you like teaching?

Say again?

Do you like teaching?

Yes, I enjoy teaching. I liked computer animation so I could use that, and I eventually got round to teaching computer graphics, although initially I taught compiling techniques and then translator writing systems. Eventually went on to computer graphics and landed up on the World Wide Web, so that took me right up to 2000, basically, at Brunel.

Now, in 1979 we've got the introduction of Ada, a highly classed, shall we say, programming language. Didn't go very far, did it, Ada? Do you know why?

I think because it just didn't take off. It was a reasonably complicated language and I think it was, it was not going to take over from Fortran because Fortran was what everyone was using for large computations and it wasn't that different. So I think Pascal was more likely to take over from ALGOL than Ada was as a result.

In the same year, in '79, we've got something that did become very popular and made a basis for a lot of software, we've got VisiCalc.

Don't remember using that at all.

Now, where were you then in '79?

Well, in '79 I start at – just backing up slightly – it was sort of two years before both the, or just before both the head of Rutherford Lab, Godfrey Stafford was going to retire and Bill Walkinshaw who was running the Rutherford computing division was also retiring, and so it was an obvious time to try and merge the two computing divisions, the old Atlas one and the Rutherford computing one into one. And so that was a decision and for about a few months at the end of the Atlas Computing Division life I took over running that, and Geoff Manning, who was going to be the new director probably of Rutherford, moved back to his original job. And then I spent about a few months talking with Bill Walkinshaw, he telling me what he did and I telling him what I did, and eventually we had everything in that SRC, (a) if it's going up a level in terms of jobs you have to have an interview and other people could apply, and luckily I managed to convince them that I was the right guy. So I took over all of the computing at Rutherford, basically, for a period, four or five years, I think it was. We'd moved all of the computer equipment into the Atlas building by then, so the mainframes consisted of two oldish 360/195s front-ended by an IBM 3032, which allowed users to organise jobs, resubmit them and then to the two batch 195s. So that was one side of it and I guess the other side was that the distributed computing systems programme had started, run by SERC, and that was sort of still looking, as the ICF was, that we would gradually move away from Primes for interactive computing into single user, personal computers, as you'd call them today, but in those days the idea was to call them single user systems. And so that was the

other thread that (a) try and sort out whether there was such a thing as a single user system, and secondly, replace two old 360/195s by something else.

[01:30:11]

And going for the mainframe bit first, it was pretty obvious that we needed to get more computing power and the top of the range at the time for IBM was their 3081, which did run an operating system on top of the operating system, effectively, called VM, so you could really get VM to run two operating systems at the same time so it, very flexible, and the 3032 did the same. So there was a mechanism whereby you could gradually shuffle two 195s, a 3032 and a 3081 so that you went from the old 3032s… old 360/195s into what would be ideal would be a 3081 sitting in front of another 3081, one doing the front-end tasks and one doing the back. And the other thing that was needed was a mass storage system so that effectively you could run all of the long jobs that needed a lot of data processing through the Masstore and therefore sort of offload, you loaded up the Masstore with the magnetic tapes for a week and that meant that we could reduce the number of operators required as well. And the only problem was that there was only enough money for one 3081, so eventually we came to the decision that we'd have the 3032 and the 3081 and after a lot of manipulation the 195s eventually vanished and we got to that stage, which wasn't ideal. We'd been talking to ICL for a while still after the 6A and they were trying to get into bed with Fujitsu, mainly so that they could get at their VLSI design stuff, and they came up with an idea that they would rename the Fujitsu FACOM machine to an Atlas 10, which was ten times an Atlas, why didn't we ask DTI if that was a good idea. And DTI were very enthusiastic, the money was found, we got a good discount on it, and so it landed up with a 3081 in front of an Atlas 10 and also a mass storage system that gave us the ability to effectively keep most of the processing that we needed in terms of tapes and that online for a week, which meant that the number of operators that we needed went down by one or two per shift, which is probably ten man years. So that was the mainframe side of it.

[01:33:30]

Just pause for a minute. The Atlas 10, made by Fujitsu, was IBM compatible, was it not?

Yes, yes, they were both IBM compatible. They both ran a VM system, so one of the things we did actually was to run a Unix system as well. Unix system had been developed by Amdahl who were making sort of IBM compatible machines, and we ran that on both the machines so the 3081 could run Unix programs as could the Fujitsu machine, which was quite nice. And the Fujitsu machine was probably the fastest thing around, apart from Vector processors. And so it was a significant advance in terms of computing power for the department, and it served them well for a few years, a result.

[01:34:29]

It was probably the only Atlas 10 sold in the UK wasn't it?

I think so, yes. Yes, ICL were in a real mess at the time in the sense that they had been, they'd gobbled up English Electric and also Elliotts and they were having to move people around, trying to make some sense out of a number of different systems they were selling and it never really came to… they were still in that stage where they didn't know where everyone should be and it was a problem for them and no one took real pleasure in Fujitsu other than we did, so…

Did IBM come back and make a counter offer?

At the time, no, they were happy with the arrangement of the 3081 in front of a… They came out with a totally different offer, which was why not sell the 3081. We'd had the 3081 for two years, which meant that we could actually sell the 3081. And they said – they had a funny arrangement, IBM, that they only changed the exchange rate sort of once a year and so you worked out that if you sold a 3081 to Volkswagen, we could get more money or as much money as we needed to actually buy another 3081, a later one which was even faster. And so IBM arranged the deal for us, so it

worked out quite well. Over a weekend we got rid of a 3081 and got a newer 3081, which was larger and faster and no one noticed, other than everything went a bit better. So it was a good deal.

They were made in America, weren't they, the 3081s?

Say again?

They were made in America?

Yes, yes.

Poughkeepsie, probably.

Yes. They were made up in Poughkeepsie, and I think we went actually and watched some at one stage to see them being developed and that. Quite amusing because people were stealing bits from an assembly line and people, if they were short of a bit were stealing off another, which they had to sort out. [laughs] But yes, it was very impressive, the IBM plant at that stage.

They're not, IBM I don't think is given enough credit for some of its operating systems, because VM really was an extremely powerful virtual machine operating system.

Yes, it really made life a lot easier in terms of you could effectively go from one version of the operating system to the next in a way where you ran up the new system while still running the old system, and then eventually switch over, yeah. It was very useful, and I think, yeah, and IBM did a really good job on it, it worked well. The fact we could run ALGOL meant that it really gave the whole Unix community the ability to run much larger jobs than they would ever anticipated being on a basic Unix system.

The magnetic tape you used, was this one of IBM's magnetic tape stores?

Masstore, no. Masstore was a separate company which had a load of cartridges and the cartridges all – trying to think how many of them – but it was a lot and they were quite small cartridges, but it was about 110 gigabytes, I think, in total size, so it was quite large amount of mag tapes could be put onto the Masstore, and it was very fast and it whisked around moving these cassettes in and out at tremendous speed. Masstore was a small company just below San Francisco, that part of the country.

It cascaded it, did it? It went from tape to disk, from disk to memory?

Yeah, it was little cartridge tapes and you put the mag tapes, the old IBM mag tapes, say once a week, and you filled up the Masstore with the ones that the particle physicists or the astronomers wanted to process in the week, and then effectively you didn't have any operator intervention necessary for that loaded mag tapes for those particular jobs. And there were a lot of them. A lot of tapes came back each week from CERN. And a lot of tapes came back from the satellites we'd put up and they all got processed locally, yeah.

[01:39:56]

So you were there five years, were you?

Yeah, it was about that long that I was running the… And the other main thing was the single user system side.

What did you choose for your single user system?

We thought that the ICF and the DCS programmes had come to the conclusion that if the Primes could get a bit smaller you could eventually afford to have a computer of quite high power as a personal computer on your desk effectively. Trouble is, no one could do such a thing, and so that was the problem. But they reckoned that the size of things like the Primes was going down by the year, you know, whatever you could process in one year would be much, much more the next year, so it was going to get there and the matter was when. And around the beginning of '79 it was clear that there was the likelihood to be fairly soon. Rumours kept coming around there was a

small company in Pittsburgh that was going to do it. And the rumours were just rumours and so not everyone thought much of it. Eventually there was- I'd been to a meeting in France about methodology of interaction, and basically there there was all the wise and good men who could do good interactive computing, like people from Xerox PARC, CMU, Toronto and elsewhere. And everyone was saying the future is Three Rivers in Pittsburgh's going to be the first personal computer. There'd been some attempts by IBM and Apple, but it was so Mickey Mouse compared with what people wanted, it wasn't good enough. We started looking at the PERQ and it clearly was impressive, it was, it had an A4 display which looked as good as a sheet of printed paper, and you could move stuff around on it at an incredible speed, and it had a largish disk, quite a bit of memory, and it ran beautifully. The reason I think behind that was that Brian Rosen – Three Rivers had been in existence for a few years as a company – Brian Rosen and a guy called Stan Kriz, I think his name was, the two of them had developed some high speed really nice displays. And Brian had left to join Xerox PARC and he put most of what they'd done in Pittsburgh onto the PARC Alto machines and suggested to Xerox that they ought to start up a company and run it, and Xerox said no, that's not our business, we don't want to. So Brian Rosen decided he'd go back to Pittsburgh and we'll do it there. It was a really small company, about ten people, I think, that sort of size, and they had a little facility and a bit of a… But what was coming out was that Toronto had put in a letter of intent, CMU had put a letter of intent in for ten PERQs, that was the name of the machine – PERQ – and clearly we ought to be looking at it again. And everyone was saying well, buy one, buy one. And eventually we got round to it, we saw it at a SIGGRAPH, they showed off a handmade version effectively of what they were going to manufacture. So none of the three boards in the PERQ were there, but you couldn't get near the stall, people were just ogling at this major change to life, kind of thing. And we came back and decided we'd buy one. We'll get it as quickly as possible. So we got the, we went to the Embassy in Washington and said well, can you buy it, because it was going to take us time to get through it, and they said, yeah, we'll buy it for you. And they bought it and they put it in a crate, eventually it arrived at Rutherford. Meanwhile we had been talking to ICL about small systems and whether they…

[01:45:34]

Sorry, how much did it cost?

It was about 20,000 dollars.

Yeah, okay.

It was that sort of price, the prices kept changing, which is why I'm sort of… But 20,000, £20,000, it's that sort of area that we're talking about. So quite a lot of money for a single user system and not everyone would be able to afford it, but it really was impressive, what was being shown. So, we got the machine, it eventually arrived, it had been repacked into a big sturdy wooden case, which meant that it was wobbling up and down inside and was bust by the time it arrived in Rutherford. And luckily, as I said, ICL had been looking at the possibility, and Charlie Portman from ICL who'd got a system, I think, called Project Little that they were working on, which was effectively something similar. He reckoned they were two years away from being able to do anything similar. And so we got Charlie to come down while we were unpacking this thing and he looked at it, worked out what was wrong, one of the boards had got bust at the corner, kind of thing. And basically said, well, I'll take it away and mend it for you, which he did. Came back the next morning with it all working, and he said, and by the way, I've now put together the cost of all the bits and pieces and worked out how much ICL could sell it for and make it for. So as a result we sort of came up with why don't you do something about it. So I wrote a letter to Ninian Eadie saying, well, either buy it through us or come out with an agreement where you manufacture outside the USA and sell worldwide. So that was a sort of solution and, I don't know, it's probably two years nearly of negotiation around whether or not this would work and who would fund it and everything. But I'll cut a long story short to say that eventually it was decided that (a) ICL would make them, (b) they would have the right to sell them worldwide and SERC agreed to buy 200 of them, which is quite a lot. And that's what happened, effectively. Only problem was that the PERQ ran POS, which was a local operating system, based on Pascal, and basically it was a soft machine, whereby you could actually define the instructions set at the machine itself. So Pascal p-Codes is the basic machine code of the machine.

And so it worked very efficiently and ran very well and everyone was saying we ought to have Unix on it, so between us, after a couple of years of effort and that, Unix eventually appeared on the- sorry, about 40 or 60 of the PERQs that we put out initially were running a POS operating system and eventually after a couple of years we had 200 PERQs and most of them were running Unix, but it took a lot of effort. I'll not go into it, because it will take about three weeks to go through the detail, but that's probably good enough.

[01:49:35]

And the only trouble was that once PERQs started selling from Three Rivers, everyone got on the bandwagon, so Bill Poduska was head of Prime, left Prime and started up a new company called Apollo, which was effectively putting out, you know, the old Prime operating system onto a single… And the display was quite good, not as good as the one that Three Rivers had, but it was okay. And then some of the PDP-11 people who were producing newer and different versions of Unix thought, well, we can put a decent display on the front of a PDP-11 effectively, and so someone like Sun Microsystems got into it. And as always, once you get sort of a bandwagon effect from the US, the original one doesn't get, you know, get the speed of take-up that you would want and so eventually ICL stopped selling PERQs and it was probably a glorious failure in some sense, but it did mean that at least two, probably 100 or more groups of people in the UK knew what a personal computer was and how to put them together.

I'm always surprised the Digital Equipment Corporation didn't step firmly into that market.

Say again?

I was surprised that Digital Equipment Corporation did not step firmly into that market.

They kept thinking about it and I don't know why they never did it, because they were the obvious ones in some sense. They had, both had displays which were quite good

and they had the machines, and the machines were getting smaller. They came out with some really small things but they never sold particularly well. I think it was just once another company could (a) upgrade Unix and put out a PC version that they gave up and carried on with their existing, yeah. But I think it was strange as well, I would have expected them to have done it, but they didn't.

It was natural for them, I thought, but I don't know why they didn't do it.

I think you're right and I thought so too at the time, but they weren't interested. They weren't interested in Unix, to be honest, it wasn't the next generation when people were making, you know, other companies were doing the same and doing DEC compatible systems like Systime did that they even were really feeling that they should run Unix. So they stopped running Unix on their systems, you know, it never really took off as a worthwhile operating system.

[01:53:00]

We're into the early 1980s now, aren't we?

Yeah, we're probably up to about, I don't know…

Now, 1981 and the Japanese launched the Fifth Generation Computer Project and there was a response in this country in 1982. The head of technology from BT, Alvey, sets up the committee and they establish the Alvey Project, which has four streams to it: software engineering; intelligence knowledge-based systems; man-machine interfaces; and down on the hardware level, BLSI. What did you think of Alvey?

I think it was the right approach and they got the areas about right as well. I'm sort of slightly biased because we got involved with the Roberts Panel that was an earlier attempt at it. But it was clear that quality of software was needed better, it was clear that knowledge-based systems were appearing and there was a future there, just as like it is today that there's a future for AI somewhere that most people believe. It was clear from the personal computing side that interaction between man and machine was going to change quite dramatically, and so that was also an area. And effectively

there was no great standardisation or understanding of what was the best way to do things, there were a lot of possibilities. Local Area Networks, you could have ethernets or Cambridge rings, and so on. So yeah, it was the right thing to do. Probably better than to try and be part of the Japanese Fifth Generation, so it was a major attempt at bringing stuff forward as much as possible.

Reading between the lines and also having interviewed some people about it, Alvey seemed to be as a project rather chaotic bureaucratically.

I didn't get involved in it that much at that level. I mean by the time it was decided to be done it was going to be done through DTI, so the basis for the company was to put it all together from there. So effectively the heads of each of the departments, the four that you mentioned, were in Millbank in London and that was the heart of it. DTI put in a lot of the funding and so they wanted control. SERC was well behind it and so said they would do as much as they could to help. And I think Alvey probably worked pretty well, but most of it, so Informatics, we eventually split after a few years, as you probably gathered, from one computing department back to two, and I was then running a department called Informatics and the mainframe side went off to a different one which I didn't get involved in after that. But we were sort of asked to do the infrastructure for the Alvey Project, which was, yeah, most of the Alvey stuff was either going to be done in industry with research in the universities, that was the sort of aim, and the aim was that they really ought to have a standard infrastructure and the decision was made that it should be Unix and so on, and so it was, as you'd expect. The only thing is they had to do a sort of an Alvey ICF effectively, so our job, we landed up by buying a load of machines for them. GEC ones running Unix and Systime ones running Unix and they formed the basis for the projects that ran, and I think there was about ten of those that we bought and installed and put them on the network and everything. We'd done it all for the ICF, we'd done it all for Starlink for the astronomers, and now we did it again for the Alvey side. And also, they were interested in having some of our staff help out, so we had probably projects or actually just being the sort of dogsbody who went round and looked at the researchers after they'd been awarded, we did that side of it. So yeah, we got quite heavily involved at an arm's-length with Alvey with the staff involved in Informatics who were working on Alvey, spent most of their time wandering up and down to London

to be involved in the projects. We got involved in, from the research side, Informatics was trying to get enough funding to pay for the staff it had got, which was only about 40 at the beginning, so it was quite small. But we did some activities in all three of the four areas, we didn't do anything on the VLSI side, which was sort of Technology Department's area. We did the other three. So, as well as putting the machines in, we ran a sort of software, a hardware thing which, you know, allowed us to put together things like tape decks to put onto people's machines so they could talk to each other, anything, lots of little bits and pieces. Cambridge Rings and ethernet, stuff at that end, so all the side things to go around the researchers main facility we put in. And that worked out pretty well, so we had quite a bit of sort of manpower that was involved in one way or another with it. So yeah, we got highly involved in it. I personally didn't get much involved and the people who worked for me basically reported to someone in London, so I used to go and talk to them every now again to find out what on earth was going on, but at that level. But we ran grants for them in IKBS area, the MMI area, and software engineering, so all three we were involved in.

[02:00:41]

What was your involvement with the transputer?

Sort of back off slightly, it was the DCS Programme and the Distributed Computing System Programme, one theme of it was to, was Tony Hoare's Communicating Sequential Processes, which was a sort of an academic view of how you ought to put things like Base machines together in clusters to make up something, you know, distributed computing of some kind. And there was a group at Warwick in the DCS Programme who'd shown interest. A guy called Ian Barron who used to run a company called CTL was the person who ran the DCS Programme and he saw the people at Warwick and said well, why don't we start up a company called INMOS and make a thing called the transputer, which is effectively what Tony Hoare had said was the kind of way forward, and start selling them. And that probably took about three years, so the transputer initiative came about because suddenly SERC was getting loads of people saying can you do this research and we want three of these transputer things, and eventually they got to the stage where they're going to have a, why don't we have a transputer initiative and try and get them all to do different

things rather than all try and do the same thing. So that was the transputer initiative. And Ian Barron was at the Warwick staff, Whitby-Strevens and May, I think it was, there were two or three of them also who went to INMOS, and so the transputer came. So there was this little computer on a chip which you could put together into sets and the whole of the transputer initiative was concerned with what you could do with a set of transputers, basically. And it got good funding and you could, you had a great big pool of transputers, so you could sort of say, do a little experiment with four transputers and when it was running, then you'd say, well, I need to show it off by changing the four to 64, or 24 and so we had a great pool and the transputers would be loaned out and brought back and stuff like that. And that worked pretty well. And yeah, it lasted for a few years and eventually SERC and DTI decided it had enough of- but everyone else in Europe decided they were still interested in transputers, so the actual transputer conferences, I think, went on for another three years afterwards. So after the transputer initiative it survived for, I think, really two or three years. One thing I do remember is that we had a driverless car in one of the, which was run basically by having a transputer where the normal back-facing mirror is, and had it look at the road signs and navigate its way to, I think it was either Dusseldorf conference centre or something on the roads of Germany, and that was quite early on compared with local activities now.

[02:05:00]

Mm, yeah. Absolutely. You remained head of the Informatics Department through the eighties, yes?

Yeah, Informatics lasted quite a while. We originally thought it would only last for five years, but we inherited the magnet design people from technology department, so that all of the engineering board activities were in one department. So the 40 in the beginning of Informatics eventually went up to about 100 and we were running that pretty well and that worked when we got involved with sort of magnet design finite elements. And VSLI design with some ESPRIT projects, and it was all quite good and worked out pretty well. But, the big thing that really ruined Informatics was after ICF had finished, the Engineering Board had a thing called EASE, which was Engineering Applications Support Environment – so instead of telling universities

what they should buy, we turned into a sort of 'Which?' outfit looking at various things and reporting on them. It was less exciting for us, but we got a load of odd requests for stuff. And that went on for a couple of years, I would guess, or a bit more. And then all of a sudden Engineering Board was short of money and so they sort of, next year they said the 50 man years that we were currently providing you with will be 10 man years, so we had to remove 40 people from… Didn't make any of them redundant. Rutherford basically stopped anyone recruiting any computer person and they first had to justify why the person in Informatics wasn't good enough for the job they wanted and so quite a few people became parts of other parts of Rutherford as a result and Informatics got a lot smaller. That's what happened over the next period, which was, it was quite interesting, but not very fulfilling in some ways. But yeah, we did that until, I'm trying to think when we actually finished it, but eventually it was decided that Informatics couldn't last any longer, so I think it was about 10 or 12 years and we then merged the two departments back together again, so at that stage I was without a job and we basically went on to, I was doing more things by then outside of the Rutherford as well on the side. We'd joined an organisation called ERCIM and that was the starting point.

What does that stand for?

The European Consortium for Informatics and Mathematics. I mean effectively CWI was the sort of leading government funded organisation in Holland, Netherlands. GMD was the same in Germany, and INRIA was the third one in France, and the three of them had got together for a year and basically had meetings so that people working in a particular part of informatics could talk to each other and work together. And we'd had several projects with INRIA from ESPRIT round simple workstations and various bits and pieces, so we knew INRIA well and they came to us and said did you want to join ERCIM, and as being the equivalent in the UK you're not quite equivalent because you don't do just informatics or mathematics, but everything else as well, but we know you are significant in terms of that. And so we got the director and the director agreed we should join ERCIM, and so we became the fourth member.

[02:10:04]

And they then decided that as things were, they really ought to try and expand to cover Europe as a whole, so four of us, one from each of those four countries, went round seeing if there were equivalent organisations in any of the other European companies. I can't remember the order, it was something like we went to Italy and decided that the CNR Group at Pisa was the right one. We went to Greece and decided that FORT-H in Crete was the right one. Then we did the Norwegian one, which was SINTEF. SICS in Sweden, VTT in Finland. And so it went on, and Sztaki in Hungary. And eventually we'd gone out to about 10 or 12 members and the directors met once a year and at the same time there was a working group that used to come together as ERCIM working group and meet at those meetings and have a sort of, yeah, effectively a separate little conference at the same time. And by the time you've got eight to ten meetings of ten people it was quite a lavish activity over the years. And they eventually decided that they needed someone to run this that weren't the directors, and so they set up an executive committee and the first chair of that I took over. So when we got to about eight I was chairing that, and did that for a few years.

What concrete came out of that?

Well, that- what concrete came out of it, you say?

Yes, what concrete value out of it?

Well, in terms of getting involved in things, I think that the first thing is we, one of the ESPRIT rounds where half – ERCIM had at least one, if not two, partners in 50% of the awards of research grants in Europe, so it was a significant activity. The main thing it did was to try and get people to move around in Europe. So young researchers used to go from one department to another to do projects. We went to the Commission and said you are all into moving people around, why don't you fund that, well, instead of us having to fund these people wandering around Europe, why don't you do it. And they said, well, why don't you become an EEIG? And we said, what's an EEIG? And they say it's a European Economic Interest Grouping. And, what's

that? Well, he said, if there's two large companies in say, England and Germany who want to set up a start-up to do something different that neither of them do, they have the problem that, you know, they have to get a grant, whereas if they were just one company that had the backing of two companies in two different European countries, then we could count that as necessary for getting a research grant. Research grants had to have at least two members involved. So if ERCIM became an EEIG, then they could basically have this project for running all of these students and research students and running them around between three establishments with just one single grant, which would make life easier for them. So all the ERCIM partners joined the ERCIM EEIG, apart from one, I think it was. And then we had an enormous grant from the Commission which allowed us to have a, recruit a researcher for one of the many projects that we were running and if there was two of us in the same project then the guy would spend six months in one, six months in another and six months in a third. And we did that for quite a while, and it really did mean that there were quite a lot of inter-movement between the various members. And we set up working groups and that was roughly how we got into the World Wide Web as well. So that was the starting point of it. So during the time I was – if you want to go on to that, I could talk about the World Wide Web side as well.

[02:15:46]

I think you'd better because it's probably going to be quite important.

Yeah, alright. So basically the World Wide Web in the first four years didn't do very much. There were less than 500 websites worldwide. But Rutherford was 12, I think, on the list of websites, so we missed the t-shirt that had the first ten on, but otherwise were in at the beginning. And the idea was that after about four years, 1994, when it was beginning to look as though it might be something a bit more than a local particle physics thing. But Tim decided to move to the USA and to set up the World Wide Web Consortium, W3C, and the idea was to have three major posts: one in Europe, one in America and one in Japan, as the three main places. And it was clear that the European one was going to be in CERN, the American one was going to be funded by the US government at MIT, and the third one, at Keio, I think it was, in Japan was the third host. And those three hosts would effectively run the World Wide Web. So

W3C had Tim Berners-Lee in charge and CERN was going to find someone to run their end of it, and similarly Japan. And then at the last minute CERN decided that they would pull out. They were making the Large Hadron Collider, which was important for them, and this was a sort of side issue that had been dumped on them by the fact that Tim Berners-Lee had started it. So the Commission was left with trying to decide what on earth to do about it. My guess was the logical thing was to let the ERCIM EEIG do it, but they decided that was probably too adventurous for them and there was a meeting in Brussels and it decided that one of the ERCIM partners, which was INRIA, would be the European host for it. And Jeff Abramatic became the head of the W3C host in Europe and the first thing he did was to come to a meeting of the ERCIM executive committee and say can you help, we've got staff, but we've got holes in certain areas. And so us, GMD, CWI, a few others, provided early support for their activities. They had an Amaya browser which was the sort of definitive browser for the web that W3C ran and gave away to anyone who was interested, and we provided the XML support for that and also CGM. So the Computer Graphics Metafile, which was a similar sort of transfer body and an ISO standard, and we put together our CGM profile for the web, basically, which we then managed to get as an ISO standard. The only thing good about it was I wrote it in HTML, which is I think the only ISO standard that's in HTML rather than something else. It had about 400 pages if you printed it, with 800 tables. So it was probably the largest HTML page up to that point. I used to run the whole of the European web, or the UK web initially, on to my machine every morning to see if there's additional pages, it was that small. Eventually it got larger.

[02:20:28]

But as far as W3C, it runs by getting a large amount of money from major companies and a smaller amount from start-ups and universities and people like that, and the job of W3C was to set up local centres for, so the offices as they were called, one per country, and those were the way to spread the web. And lo and behold, the only condition was that the local office had to be a W3C member. So we managed to get all of the ERCIM members to join it because, W3C, so we were all members and eventually they all agreed that they would be the offices as well, so ERCIM became the offices and Rutherford became the W3C office for the UK. I spent probably two

years or more wandering around England telling them how marvellous W3C was and the world wide web and why didn't they do something about it. We got funding initially for the world wide web, for starting up the web core thing, and the next thing was the W3C Leveraging Action, W3C-LA, which lasted about three years and was, must have been '97 to 2000, something like that. And basically made, tried to gain membership in Europe. So in the UK we talked to pretty well everyone, went to the BBC and they were interested, went to BT, they were interested. Lloyds showed some interest but managed to sort of get in as a junior cost rather than the main cost by, they had a little organisation that did research for the banks and that joined instead of Lloyds itself. Mean lot, was the kind of view of them. And pharmaceutical industry was interested and joined, Reuters joined. So it was a real mixture. A lot of universities joined. And similar kind of things happened around Europe, so the amount of uptake in terms of W3C members in Europe was much better than in the USA for a while.

Why was that?

I think it was mainly because we got organised better and the fact that we had offices right across Europe without too much trouble may have worked out pretty well. So that was the involvement with ERCIM. I worked on that pretty well up until I retired in 2000, and that was the main part of my job. I retired in 2000 and at that time the head of offices in W3C had left and they needed someone, so I then went and worked for W3C, being paid by INRIA. It was alright, it worked out reasonably well. And so I wandered around opening other offices for W3C that they needed. They wanted one in an Arabic country, the reason for some of these was because of the multilingual side of things, and so- and the only requirement, they had to be a W3C member in the country and then we could define an office, so we opened one in Morocco, one in Israel, and I opened one in Australia as well in that year while I was working for them. And there was a problem that I was too old in theory to run the job at INRIA and so they decided that for the second year if I carried on I would have to go and work for MIT. And it got so complicated it was easier for me just to retire, which is what I did.

[02:25:37]

Okay. Now, with all that experience, I want to ask you some questions about what is going on now and some of the concerns people have. I'm sure you're going to have some views on this. Major concern number one, quantum computing, will this ever come to fruition, in your opinion?

Well, quantum computing will do certain things and anything that involves sort of, that you can do by sort of a lot of randomness and working out, you know, it has a use in that sort of area. I can't see it ever taking over in the mainstream, but the trouble is, having said that, if you look at some of the things going on, they look very quantum theory-ish, if you like. So the use of neural nets and stuff like that is similar in some ways. So, probably I don't think it's going to ever be mainstream. I may be wrong.

Do you think that quantum computing will pose a threat to security?

I think everything is a threat to security. I don't think they're the only ones. I can think of a lot worse, probably.

Okay. Now, what about this problem that AI systems have of hallucination?

Of what?

Hallucination.

Hallucination?

Yes. In a black box, we don't really know how they work.

Well, I think if you have an Artificial Intelligence system, right, then the first thing it must do is explain to you how it works, right? And that's the definition of an Artificial Intelligence system, that's why we call them intelligent knowledge-based systems, not AI, which stands for Artificial Idiot, I think, but… sorry. So, you know, it's a fundamental requirement and pretty well everything that is labelled AI at the

moment is not AI, in my definition of the term. And, you know, it does need to explain why it makes a decision, and the fact that they've now got multi, more than one, maybe up to four different neural nets being involved in making some judgement, unless you can explain (a) at the very little, which of the four neural nets came up with the conclusion that you do this, or you know, that you chop off your right arm or whatever the decision is, you'd need to know why. And any AI that doesn't have an 'X' for explanation in front of it is rubbish. So there's a lot of rubbish around at the moment and not much XAI, ie, AI able to explain it. To be fair to the AI people, there are some attempts at trying to get the explanation side sorted out. I mean when we were doing IKBS you did that first and then, so you know, most of the systems that we put together, you explained why they did what they did. And the same thing's needed for the current sort of set of AI so-called systems around. And once you get to XAI you could at least say that the decision we made on this or that is because (a) these two, backed up by this data suggests that you might be one of the people who are affected. You know, that needs to be put out with whatever the decision is, otherwise it's sheer lunacy.

[02:30:10]

And do you think the sheer lunacy is a threat to humankind?

Say again?

Do you think that this, as you call it, sheer lunacy, is a threat to humankind?

Yes. I think everything at the moment that's being suggested is nearly a threat. You know, it's incomprehensible the things that are going on at the moment. I just can't believe anyone would be that silly, you know, to say, (a) do this and that, dependent on something that they know nothing about, you know, it must be controlled in some way. And putting an 'X' in front of AI on everything that's allowed or talked about I think is the starting point. Everything must be explainable, and explainable to the individual involved, and if you can't then you should ban it.

Ban it?

Yeah, yeah. I mean I'm very black and white about it.

How would you ban it?

Well, make them actually, you know, if you say something is this decision, you have to explain it, and in sufficient detail and have it checked and be granted that the explanation does work. It's not difficult. It's just understanding.

Right. You've done 74 years really, of this work. 1950-53, Kayser Bondor, the lingerie company where you really started on this process, going right the way through. Now, here's my last question to you, Bob, what's the biggest mistake you've made in your career?

Biggest mistake I've made in my career? I can't think of any great ones. You could say it was the PERQ and the fact that we bought 200 of them and it didn't become the… But that wasn't just me, that was a lot of other people involved as well. That certainly didn't work out the way we anticipated. We anticipated it would be the start of ICL being a major player in the computing area and that never happened, instead they got bought up by Fujitsu. Yeah, so that's probably the biggest disaster in terms of long-term effect that I was involved with.

And Fujitsu brought us Horizon.

Say again?

Fujitsu brought us Horizon.

Yes, yes. Well, I'm not sure whether they did or whether it was ICL. Everyone keeps talking about, you know, these people in offices in Bracknell and I think, well, that was the main ICL organisation. I guess, you know, a lot of the people who are working for Fujitsu, probably some of them at least must have worked for ICL prior to that. And I think the number of people involved in Fujitsu on those particular projects was quite small, I mean we're talking about projects with maybe eight or ten

people and my guess is that Fujitsu will have thousands of people involved. I mean I visited Fujitsu when we were trying to buy the Atlas 10 and all the people I met in Fujitsu in Japan were really good. So, you know, when everyone blames Fujitsu they should blame the bit of Fujitsu that built that particular system. And, you know, there was some suggestion that the people involved were not all first class and that probably is true, but then that's probably true of nearly every project in the UK.

[02:34:55]

We do seem to have, however, in terms of many projects in the UK – I said that was the last question, it's not going to be the last question – we do seem to, Bob, have a propensity particularly in public sector projects to make a really big hash of it. I'm thinking of patient records in the NHS, I'm thinking of Horizon. Why?

Why? I guess you could argue it's a software engineering problem, still, you know. If the things were tested to, and made it absolutely certain that they did what they said they were doing, then you wouldn't have the problem. And has software engineering got better or worse? The answer is almost certainly it's got worse, and the reason probably is because 90% of the code that's running in the world is probably in JavaScript. You know, and that's made up of, normally each JavaScript program's got about ten different libraries that it's running, all incompatible with each other, and the stuff goes out and around, and why any of it works is a miracle. I notice the US government has started moving back and saying in our systems only Vanilla JavaScript will be used. So use of libraries would be a major assistance if you got rid of them all, and that will never happen.

Is there anything else you want to say in contribution to the Archive?

No, I think you've done most of what I've done up until I've retired. Since retiring I tend to spend time looking at world wide web. So I'm still a world wide web addict. But other than that, no, I think I've covered most of the stuff.

Well, I hope you feed your addiction. For once I hope a man feeds his addiction. Thank you very much for your really excellent contribution to the Archives, thank you very much, Bob Hopgood.

[end of recording]