Capturing the Past, Inspiring the Future

Professor Brian Randall

Interviewed by

Troy Astarte

6th December 2018

At the

Urban Sciences Building,
Newcastle University,

Hello, my name is Troy Astarte and I'm recording an interview on behalf of the Archives of Information Technology. The date is Thursday, 6th December 2018 and we're in the Urban Sciences Building at Newcastle University. Today I am interviewing Emeritus Professor of Computing Science, Brian Randell. Brian has had a long and illustrious career in computing, starting work at English Electrics Atomic Power Division in 1957, after graduating from Imperial College of Science and Technology. Brian worked on various programming tasks, including the creation of the EASICODE System for the DEUCE computer and an ALGOL 60 compiler for the KDF9 machine, the latter of which was the subject of his book, Algol 60 Implementation*, along with Lawford Russell, which was one of the first books on compilers. Following English Electric, Brian joined IBM Research at Yorktown Heights in 1964, where he worked on computer and systems design. After five years at IBM, Brian became Professor of Computing Science at the newly opened Department of Computing Science at Newcastle University, where he has remained since. In that time Brian's research has covered a number of topics in computing, mostly in dependability and reliability, notably recovery blocks and distributed secure systems. Brian also has published research in the history of computing, including the book,* The Origins of Digital Computers*, and the uncovering of the codebreaking effort at Bletchley Park. So, lots to talk about. Good afternoon, Brian.*

Could I start by correcting one thing?

Of course.

When I joined Newcastle it was not the case that it was a newly opened department, rather it already had quite an illustrious career, in that it started, computing started at Newcastle in, variously identified as 1956 and 1957, when a Pegasus computer was installed here. And very soon the teaching started, initially just programming, I suppose. This was the computing laboratory. The computing laboratory combined both an academic department and a computing service and that had grown really quite substantially by the time I joined in 1968- '69, sorry.

[0:02:31]
Okay, alright. Well, I think we'll get to that bit a little bit later, but thank you for the correction. Important to get those things right. So, let's begin by asking where and when you were born?

In 1936 in Cardiff, in south Wales.

Okay. And where were you educated as a child?

I was educated in Cardiff at a, first of all, an elementary school, and then Cathays High School. That was a boys' school, there was a girls' school next door, and that was a school that was a few hundred yards from where we lived.

Okay. What was your time at school like? What did you enjoy studying and what did you do well at?

I think I went through school in a bit of a dream in that I don't remember it that well and such recollections as I have are all basically really quite pleasant. But, for example, I remember turning up to school one day to be surprised that there was an exam. Apparently everybody else knew, but I didn't. [laughs]

Like a nightmare, almost.

At school, I suppose I was one of the more industrious pupils. I think I was one of the first to be wearing glasses, for example. And certainly I was very keen on reading, more so than most of my friends.

Did you read mostly fiction or non-fiction?

Well, the first reading, the first reading that I can remember was children's comics, but the comics that had a lot of writing in them, not the ones that were all pictures. And I differed from a number of my friends in that way. And the other thing I remember was really rather different, was that I kept all the comics and I read them and reread them. Then when it came to books, I certainly was keen on the reading of

books and at some time in high school the set book amazingly was *The Cloister and the Hearth*. I can't remember the author's name now, it's a classic and it had many hundred pages and I think I was the only one in the class who read it through to the end, but I enjoyed it. So I think I was a bit distinguished from the others in that way. I enjoyed things, I wasn't particularly good at sport, perhaps because of wearing glasses. Playing rugby in glasses is worse than playing it without glasses if you need glasses.

[0:05:48]
Okay. Do you think that there was any particularly important influence on you from that time? Any teachers you remember or any particular friendships?

Certainly there was a mathematics teacher who had quite a big effect. What happened was that when I had done my- I think what happened was that there had been a vague plan that I was going to go on in languages, but when the ordinary level results came out it was found, or it was noticed, that I'd really done quite well in maths and science as well and the teachers essentially persuaded me to switch. I'd already dropped biology probably at the age of about fourteen. So in some sense I was already moving away from science and they switched me back. And for A levels I did pure maths, applied maths and physics, which was a very specialised A level. I'm not sure that was a very good idea.

Did you feel you…

But in languages earlier, I had been learning French and Spanish. To my regret, it was my father's decision that I should take Spanish rather than Welsh. I've always regretted that since.

Have you since learnt to speak Welsh?

No, I'm afraid not.

That's a shame. Do you regret that you were in some ways convinced to change your specialisation? I suppose it might have resulted in quite a different career.

No, I wouldn't say that. I do recall that I was rather turned off history, which is very surprising, given my later interest in it. And so I think that was the choice of topics in history then. Doing pure maths, it probably would have been better if I'd been in a position to do physics, maths and chemistry. But I wasn't. Mathematics was much tougher at Imperial College where I went to, than at school, so I was very glad to take refuge in computers.

[0:08:27]
Okay. Well, we'll move on to Imperial College in a moment, but just briefly you mentioned your father there having an influence on the languages you chose. Can you tell me a little about your parents, what were their occupations?

My father had his own little business. He was selling furniture, he was what's called a credit salesman. So he would sell furniture and all sorts of other things up around the Rhondda Valleys and the like, travelling to the people, he would deliver things to them and then he would collect weekly payments from them. And I still have very clear memories of him each night coming and sitting at his desk and bringing his books up to date, in an incredibly neat hand. He'd had an education, I guess after school, which was a sort of a one or two-year, I'm guessing, business training. And that business training had, amongst other things, made him extremely proficient at things to do with accounting and so on. And had taught him an incredibly legible handwriting. Neither of those things passed on to me at all.

And what about your mother?

My mother, well, they met, I understand, when I think they were both involved somehow with the Co-operative Society and I think my mother must have been working there, but I'm just guessing at that. But in the war my mother, incredibly well, ran the business, because my father was taken off to the RAF. So she did the whole thing for five years, which was really quite amazing. My grandmother came to stay with us and I suppose she brought me up as much as anybody. After the war my mother essentially handed the business back. The war I think was a considerable strain to her, so certainly she had a period of ill health afterwards. But basically she

was an extremely sporty character. I was told that she had had a try-out, I think for table tennis for Wales, or something. And I was told stories about her riding her brother's motorbike and so on, all at a time when this was not the done thing. And certainly she, for example, was very keen that I learnt to swim and insisted that I do. Yeah, I think yes, I guess the other thing, I'm gradually remembering that I think I was told that my parents really got to know each other because they were playing in mixed hockey teams. So very sporting, my mother.

[0:12:09]
And did your parents have an influence on your choice to go into mathematics when you decided that was what you would study?

I don't think so. Well, I certainly don't remember much in the way of discussion, and certainly nothing in the way of disagreement about that. I was, I suppose I was the first one in our extended family to go to university. There used to be a saying in Wales that one of the most important exports of Wales, or the most important exports from Wales were preachers, teachers and water. And certainly with my father with all of his knowledge of the Rhondda Valleys and so on, was absolutely sure that I was going to go nowhere near the mines or anything like that.

Okay. Alright.

So when I did well enough in the A levels, it was then suggested I do a further year to get a state scholarship, and the like. I don't recall any disagreement from their point of view, they were quite supportive. And I got the state scholarship, but my father's income was, I assume, just sufficient for it only to pay my fees, not to cover living costs. So, and I must say, I'm eternally grateful. What he unflinchingly announced to me was that he would pay me the full amount that the state scholarship would pay.

Oh, very generous.

Yeah.

[0:14:01]
Okay. So when you went to university then, at Imperial College of Science and Technology, University of London, as it was then, you said you studied mathematics. Which particular parts of what you learnt did you most enjoy? Was there anything you felt that influenced your future career there?

I think learning a bit about computers. At Imperial College there had been work on the building of a computer, or rather a pair of computers. This was by Tocher and Michaelson, Sidney Michaelson was later at the University of Edinburgh. But those two individuals had built a relay computer which was, got known as ICCE I – Imperial College Computing Engine number one – and I'm pretty sure that was all relays. And when I was there, they had been building a second machine, ICCE II, and that project I think was at a halt then. I learnt then or soon afterwards that the famous PMS Blackett, one of the most notable people at Imperial at that time, was somebody who laid down the notion that a maths department is not the place in which a computer should be built. So that got abandoned. And I'm not sure when the work on it stopped, but certainly Sid Michaelson was very interested in computers and I was shown the machine briefly, and he arranged that I go on a programming course. Probably the first one held at what is now City University, but was then Northampton Polytechnic. I think I've got that right. It was in Northampton Square in London.

Do you remember what year that was?

That has to be, I would guess, 1956. And then I did a third year project and the third year project was basically that of writing a program in linear algebra, I'm not quite sure what it was now.

[0:16:45]
Okay. So you got quite quickly involved with the computing that was going around then. Was there something about that that particularly interested you?

I found computers fascinating and interestingly enough, I think I was the only one in my class who did. I think there were about thirty people in the class and I don't recall

anybody else being involved, either in terms of a project or going on courses or anything, other than me.

Okay. You mentioned your other classmates there. Was it uncommon to be on a scholarship? Do you feel that made you at all different from your other classmates?

No, quite the reverse, I think. To go to Imperial College with good A level grades on a state scholarship, that was the minimum. The mathematics teacher was very keen that I should try and get into Imperial College. Oh, that was another reason for doing another year, because there was an exam to get in Imperial, just as there was an exam to get into Oxford or Cambridge. But he was quite strongly against Oxford and Cambridge, I'm not quite sure why, and very keen that I should try Imperial. And it worked.

[0:18:19]
You mentioned in your biographical information that one thing that sticks in your mind is the Huxley Building. Can you tell me what it is about that that you particularly remember?

Well, the Huxley Building was a quite tall building. Seen from the outside it is rather obviously a part of the Victoria & Albert Museum complex. So a very ornate Victorian stone and brick building. It's built round a central core, so - a central open core – so you could walk upstairs and around balconies and so on for, I don't know how many, five or six storeys. And at each of those you could look down over a balcony wall and see straight down to the bottom. And one of the little experiments that I think I, and certainly others tried was what happened if you dropped something like a squashy pear from the top, it essentially completely vaporised. There were, as a building it was quite a strange one in that there were far more people in it than you realised. Only when there was a fire drill once was I amazed how many people came out of it. Maths I think was the only thing in the building, that I knew of. And my understanding is that it wasn't very long afterwards before Imperial gave up on that building. And I think it became part of the V&A. Quite why it wasn't part of the… V&A at the time, I don't know.

[0:20:16]
Okay. So, did you meet anyone at university that you had a particularly good relationship with or that you feel was influential to the direction of your future career? I suppose Sid Michaelson.

Well, there was another person in class. I'm not sure that I'd had very much contact with him during my time at Imperial, but it turned out that we were both offered jobs at English Electric. In each case the jobs that we were offered came with deferment from National Service. Indeed, I recall a visit to, I guess, Imperial as a whole, but certainly a visit with a talk by a military recruiter and the only person who at the end of that indicated an interest in going voluntarily into the services was somebody who later failed the medical. But it was rather typical then for people who didn't want to spend two years on square bashing on National Service to seek out and obtain employment in a company whose work was regarded as of national, protected in some way, national service. And so atomic power development was distinctly that and I and Mike Kelly were both offered jobs at English Electric, and Mike, who was just getting married, suggested that if I came as a paid lodger with them, that would enable them to afford the mortgage. So for, I'm not sure how many years, I was a lodger with Mike and Ingrid Kelly. And he and I worked very closely together. It was great. And I've kept in touch with them ever since, and indeed they visited here not very long ago.

Very nice. A very long friendship.

Oh yes.

So aside from being able to get deferment from National Service, was there any other particular reason that you went with English Electric? Was it the first place you'd been offered a job, did you have other options?

I don't remember.

[0:23:08]
Okay. So you're working in the atomic power division and you ended up working with the computing department. Is that because of the skills and experience you had picked up at University?

Presumably so. Certainly I was employed there as a programmer to – both of us were, as programmers – to write nuclear reactor codes on DEUCE.

Sounds like a big responsibility, writing reactor codes. Did you feel the weight of that at the time?

No, we took computers really very casually. I think we both regarded computing as great fun and the computer as our toy. So that the notion of the weight of responsibility, never thought that at all, I'm afraid.

Okay.

In those days you typically would write a computer program and then do hand calculations to try and prove that your program is generating the right answer, the right answers. I used to get the impression, or at least give the impression, that I used the computer to debug my hand calculations, rather than the other way round.

Okay. So, you worked with Mike Kelly for a period of time at English Electric, and you've spoken before about your high regard for him and his ability as a programmer. Is there anyone else at English Electric you particularly got along with or worked well with?

I can remember quite a few names from English Electric, either of people that I worked more or less, rather less, closely and closely with, but who typically I've had contact with since. And then after Mike left and joined IBM, I was made head of a little section, and so there were certain people there who were working under my direction. Again, I can remember all of those pretty well.

What was your responsibility as head of the section?

It was to do with the writing of what we called automatic programs, or compilers, if you like, for the upcoming KDF9 computer.

Okay. So is there anyone you didn't particularly get along with, after you were working at English Electric?

My first boss there, he was okay, certainly a nice guy, but more of a computational physicist than a computer person. But then he was replaced by somebody who I had less respect for and I think he got on better with me than I got on with him, if you see what I mean. And so I felt, in the case of both of these bosses, either we or I managed more in spite of them than with them. Probably an experience that has had a, what some people would regard a bad effect on me, for the entire rest of my career.

What's that, a disregard for authority?

An appropriate scepticism, shall I say.

Of course, okay.

That is my, I claim the word arrogance or something like that.

Surely not. Okay, I don't want to go into too much detail about English Electric stuff, because we talked about that in a previous interview. I've just mentioned briefly your book with Lawford Russell, ALGOL 60 Implementation*, what kind of impact do you feel that book had?*

[0:27:54]
Well, it certainly had a big impact on my career, that's for sure. Much more than anything I realised it would have. We were writing this, this compiler, the ALGOL compiler, and somebody suggested that since we were sharing our design with others, not least because the KDF9 was delayed, and our design was first got going by somebody else, it was suggested that the various people who were sort of working with us, and in some sense ahead of us on our design, would appreciate having more

in the way of information, I guess, about the compiler. Then there was the idea of, why don't you write a book about this. And so the first thing we did was check that Edsger Dijkstra was happy about that and he wrote back very quickly saying he was very happy about it and gave us the wonderful advice that I think I've quoted often before, of saying that a book which just describes your compiler in detail will be of interest to a very limited number of people, but if you write a book which describes all of the design decisions that you considered and the rationale that you used for the choices that you made and a review in the light of what happened, of the merits of those decisions, that will make it a much more interesting book. So that's what we set out to do. We by now – I'm talking about Lawford Russell and myself – Lawford joined essentially as a replacement for Mike Kelly.

Do you remember when it was he joined? Ninety-two, something like that? Sorry, '62?

That sounds about right, but I don't know. So we produced this book. We certainly were being competitive, there were a number of other ALGOL compilers around. Tony Hoare worked with his wife on one for Elliott. We knew of Peter Naur's work on the, I guess the GIER computer. And the work of the Dutchman, van der Mey. And we, when we decided that we were writing a book on compilers, I suppose we assumed there was only going to be room for one of those, and we'd better get it out quickly. So we wrote it mainly in our spare time, in I think about nine months flat, working just about every evening and weekend. And it was only later that we find out that the others either weren't doing a book on their compilers, or certainly weren't working on our timescale. So, getting it out and getting it out that early, and getting out a book of the type that Dijkstra had recommended was certainly, that had a very good impact, I know that. And it meant that when I started looking to move on, the existence of that book helped. I remember I was at IBM, I'm not sure if it was when I paid a visit to IBM or soon after I joined IBM, but one of the people who I met was one of the vice-presidents and he was carrying a copy of my book, our book, in his hands when he arrived.

And that would have been in 1964 or so?

Yes, I guess '64/65.

[0:32:15]
So, when you did go on to work for IBM you had stopped working on compilers…

Well, I went on to IBM, I joined IBM with the express wish to not work on compilers there. Mainly because I didn't want to be bought by them just to, because of my ALGOL compiling knowledge.

Okay, I see. Was there any other particular reason you felt you wanted to move on from compiler work?

Not particularly. I think IBM was such an amazing place that I guess I was thinking that to not take advantage of that and learn something new would be an awful waste. I'm guessing I thought of it that way.

So why was it you chose to work for IBM in particular then?

I was pretty overwhelmed by- well, first of all, when I was first invited out there, I refused. The idea of going to work for IBM struck me as really quite frightening, it being such a huge and successful operation compared to English Electric. When they said we understand, but why don't you come out and see us. Sorry, let me go back a paragraph. When I was first contacted by IBM and asked would I be interested in joining them, I said no. And when I said no, to my amazement they then said we quite understand, why don't you come out and see us. And so I was out there for several days and one of the people who took a considerable interest in me was John Cocke. And John Cocke, later a Turing Award winner, an amazing, very impressive character, and for example, he drove up specially with me to Poughkeepsie to show me the factory there, just so that I could see how IBM produced computers. Well, with that sort of treatment I hopefully gracefully gave in and with some qualms agreed to join them.

[0:34:56]
And what did you work on once you were working for IBM?

I joined what was called Project Y, which was a project to design what you could call a supercomputer then. A number of the people involved had been involved in the Stretch computer. Stretch had been IBM's attempt to build the world's fastest computer and it was somewhat successful, but in that part of the market they had in CDC and Seymour Cray, a very strong rival, so that Project Y was an attempt to get back into that market and to beat Cray and CDC. But this was just a research project at this stage. Mind you, there were probably about twenty people involved, when it was within the research department.

And specifically what were you doing as part of that effort?

I was mainly, the first term task I was given was to design the order code, or to help design the order code. And I worked on that with somebody called Herb Schorr – that's S-C-H-O-double R. And we were working for John Cocke, John Cocke was interested in everything, but in particular interested in compilers and optimising compilers. And Herb and I began to believe that worrying about the order code was in some sense a question of sub-optimisation. So there was a much bigger problem on the machine, which was that of making sure that data and instructions could be obtained fast enough to keep the arithmetic unit busy. So there were more problems to do with the overall design of the machine and in particular of things like operating systems and so on. We didn't get very far at that stage with John Cocke in arguing that.

So you felt that a bigger picture was necessary?

That's right.

Okay.

But then the decision was made by IBM to go for real and to transform this research into a whole development project, and so I think it was about ten of us were asked to move out to the West Coast to initiate the creation of a whole new computer development department. That was called ACS, Advance Computing Systems. All of

this was highly secret, by the way. It was pretty secret from the rest of IBM, and certainly from outside.

[0:38:07]
So, speaking generally, how did you find the switch from working for EE to IBM? Quite different companies, how did you have to adapt your working style?

Certainly it was different and I was finding my way. In retrospect I think that, well, I was joining a place where there were, I think the IBM Research Center, if I remember rightly, had something like 3,000 people of whom half had PhDs, or something like that. So in that sense it was overwhelming and I certainly didn't have a PhD and I hadn't regarded what I'd been doing at English Electric as research. But it became clear that IBM did [laughs], and it was rather more important what they thought of it than what I thought of it. So, as I found that I was being treated well and with respect, so I gained confidence - so I gained over-confidence, one might say - and soon was enjoying myself greatly.

And did you reconnect with Mike Kelly?

No. Well, yes in the sense that we visited him once, because by that time he was in the States as well. He had joined IBM Hursley – that's near Winchester – where he was involved in IBM, one of the IBM 360 development teams. He must have been one of IBM's first micro-programmers. But he moved out to IBM Federal Systems Division and I cannot remember now whether he had gone out there before I did or not.

[0:40:17]
How did you find the change when you moved over to the West Coast and the switch from the research project into the development project?

Well, I got married before, about a year before we went out to America, and so we very much went out as a married couple. And fairly soon after we got to America my wife became pregnant, so the move out to California was hurried. We were the first of the whole set of people to go out there, just so as to get there before the baby was

born. And so my first couple of weeks there were in the IBM Los Gatos laboratory in this wonderful office which belonged to an IBM Fellow, it's just that he wasn't there, so they said, you sit there for a few weeks. And gradually the others joined in. But I was out in California only for about a year. But that was a year of our first child and so on, so things were very different domestically as well.

Yeah, I imagine so. So at what point did you begin to think about questions of dependability in computing?

I think that, well, at English Electric, we certainly thought quite a bit about the practicalities of the programs that we were developing. So I – I think it was I, rather than Mike – formulated a rule about the sort of programs that we were generating. Because we were generating programs for other people to use, application programs. And for our own benefit, if nobody else's, we tried to write those programs so that they could cope with all sorts of nasty realities. Incompetent operators, incompetent users, incompetent key punch operators and so on. And we formulated a little phrase where we actually used the names of three particular individuals as being what we would regard as the way of producing. I think we called it 'complete' programs. I don't think we called them- we certainly didn't call them dependable programs, and we didn't call them fault tolerant programs, but essentially that's what we were doing. But doing it for entirely practical purposes. So, in that sense, you might say, I became interested in dependability. But…

Were you coming up with technical solutions at that time to cope with problems such as poorly punched cards and so on?

Well, one of the rules was that a program should deal sensibly with any input that was given it. It shouldn't just fail, it shouldn't just check, it should give you some information as to what the problem was, if you see what I mean. Then when later we were writing programs for the KDF9, the KDF9 had this little sixteen-word pushdown academic arithmetic unit and one of the things, one of the faults you might have in your program was by accident you leave something in the pushdown store, the nesting store this was called. And if you had some big and complicated loop, after you'd been round that a few times your program would suddenly halt, because the nesting

store overflowed, because it had all this gash in it. What we realised was there was a very simple thing you could do, which was at some appropriate point in this big loop you'd check that the nesting store was indeed empty, which was what it was supposed to be. And if it wasn't, then you flagged that up and you got an immediate indication of something going wrong, which otherwise would have gone wrong long after at some quite arbitrary point in the program, leaving you no idea as to where the problem was. So, that was an example of the sort of programming strategy, if you will, of testing and reporting sensibly. And then similarly, well even back in the days of writing EASICODE, we tried on grounds of practicality to have a system that was easy to use. When you live in the same large office as the people using your programs, then you pay a lot of attention to their problems so that they don't become your problems.

[0:46:14]
And were you allowed to continue that kind of interest at IBM, or were you looking at a higher level system design?

No, I wasn't particularly concerning myself with the reliability issues then. I guess it was more, well, I say initially it was to do with the order code, then when I went out to the West Coast, we were concerning ourselves with the architecture of the machine, and then there's the whole story of my involvement in ACS and the invention, my involvement in the invention of dynamic storage- now, what did we call it now? Dynamic instruction scheduling. So, the impact of that work on ACS was really to get interested in the whole problem of how to design systems, in that I'd been involved in the design of systems and decided that there ought to be sort of better ways of doing it. So I, when I got back to Yorktown Heights I joined Manny Lehman's project, IMP, it was called, which was one of the first multi-processor projects, certainly one of the first multi-processor research projects anywhere. And that was at a time when IBM was having troubles with OS/360 and in particular with the time sharing system, TSS/360. That was being built at a different lab, but some of my colleagues were borrowed, or spent time there and so I knew a fair bit about what was going on and what was going wrong there. And I and a colleague, Frank Zurcher, got very interested in the whole issue of how to design systems. And the work that we did together got very, very intense. We could be involved in deep conversations

that made no sense to anybody else, but where we could understand fully what was going on. And that led to our work on multi-level modelling and that work on multi-level modelling I found myself referring back to in almost anything I've done since.

Okay. So it's an idea that's really stuck with you.

Absolutely.

Okay. Well, I think we'll talk next about your move into academia, but shall we maybe have a brief pause now?

Yeah, I can have a drink.

[break in recording]

[0:49:29]
Okay. So, resuming. Tell me about your move to Newcastle University, what brought that about?

We'd been living in America for probably four years or so. When we originally went to America we talked about going there for two to three years. I was quite happy at Yorktown Heights. There had been an issue that the IMP project got cancelled and I disagreed with that decision. But that wasn't enough to cause me to leave or anything. It did cause some interesting fights between me and the management there. Herb Schorr by this time was Director of Computing and I remembered at some stage, I think we were walking out in the carpark to or from our cars, and I don't know if I actually got talking about my disagreement about the project IMP decision, possibly not, but he said something along the lines of, Brian, you don't seem to realise that the last few weeks I've been trying to arrange, trying to hint that you join my technical staff. And I said to Herb, oh, that's very kind of you, but how would I tell my friends? He didn't speak to me for weeks. So, in one sense I got on quite well with him, but in another sense, not so. So, that's by the by. Our daughter was getting – our first child – was getting close to going into school. I guess she was, she must have been, well, coming up to four or whatever. We were living in a town called

Pleasantville, which fitted its name, which had a very good school system, and we didn't have any qualms about the quality of the education there, but we were very aware that American schools produce American children. And so for all that we liked America, we didn't regard it as home, possibly me more than my wife even, and I just felt it wasn't a country that I could transfer my loyalty to. I liked it, we made very good friends, some of whom we've kept ever since. So I started thinking that I'd like to get back to Britain. I assumed this was going to be difficult and that almost certainly what I'd have to do was find some sort of one-year visiting appointment or whatever, and I started investigating those. But then, if I remember correctly, it was my wife's sister who saw that Newcastle had advertised for a second chair in computing and, if I remember correctly, I contacted Newcastle and it was arranged that I visit. I was travelling quite a lot to and from Europe at the time, through things like the ALGOL committee and so on, so I visited here and remember with great amusement sitting in Ewan Page's office and we're having this wonderfully delicate conversation, when in retrospect it's very clear that I was trying to find out the chances of my being offered the job and he was trying to find out the chances of my accepting it.

He was the director at the time?

Yes, Ewan Page. He was the person who created computing at Newcastle, absolutely wonderful character. Almost the archetypical benevolent despot, who was revered by just about all his own staff and feared by just about everybody else. Which was great. And he was very keen to build up research at Newcastle and so to appoint a research professor, he was more than keen to go on running the place and that appealed to me greatly. I was influenced also by the fact that I knew that both Tony Hoare, and I think John Buxton, had been appointed to chairs in British universities. Both of them, like me, without ever having had a doctorate. So, I didn't regard that as a necessary bar, the precedent had been set as far as I was concerned, and it was as arrogant as, if they could do it, I could do it. So the fact that I had never given a lecture course in my life, which didn't seem to faze Ewan, so it didn't faze me either.

[0:55:22]
So what was your research on when you first moved to Newcastle?

I must admit, I can't remember what I did first. What I do know is that not long, not very long after I got here, there was this enquiry from the Science Research Council as to why Newcastle had not been making applications to them. Newcastle to that date had had two fairly big, for the period, research projects, but funded by other agencies. So that letter came in one morning, by the end of the morning Jim Eve and I had got a whole plan ready and we bounced back this big application to them. They I think must have gulped somewhat, because I think we were asking for three or four computers and several research associates and so on. I can come back to what it was about, but anyway. This was all to do with reliability. And what they did was offer us a little grant first of all, which was really more travel money than anything else, to do a survey and to justify, essentially to produce a state of the art report as justification for what we were claiming ought to be done. And so that included a trip to the States and around various places in Britain. I did, I think I was involved in all of those, I'm not sure about that. But the, on several birds with one stone, that report was the required report out of this grant, it was used to buttress another research proposal and it was also used for my invited talk at the IFIP Congress. So, three for one. And we then put in the proposal again. It had probably grown bigger by this time. I'm just guessing when I say that. And we got the whole thing going and that was the start of the work on what we then called reliability. Later we switched to the term dependability. But I managed to get some very good people to join in on the work. And I think, if I look back on all of the things from then right to the beginning of my time at English Electric, you might say what I did mainly was make sure that I worked with very good people. And so that was the start of work on dependability which has carried on ever since.

[0:58:40]
Do you think the field of dependability has any particularly noteworthy success stories? What are you most proud of contributing?

I think probably the Newcastle Connection and the things associated with it, like the work on security, on load levelling and triple modular redundancy. We had a whole Lego kit of how to build systems, thanks to the Newcastle Connection. So I think that more than anything else. And we were starting to do quite well in the way of impact

and exploitation and so on. By that time there was the organisation that I'd been involved in setting up, called MARI which was a sort of a contract research organisation, and they were selling the Newcastle Connection. But then Sun Microsystems started offering free their NFS, Network File System, which was only a distributed file system, whereas what we were providing was a distributed computer system, which the files were only, as far as we were concerned, the comparatively simple part. But we couldn't compete against something being offered free by the manufacturer, so to speak. So that killed off the Newcastle Connection as a really high impact system. But it certainly gained a lot of attention. We had visitors from Bell Labs here and so on.

Do you think industry pays enough attention to questions of dependability generally?

Typically not. Typically people will start worrying about dependability only when they belatedly realise they're already depending on something. And the number of things I've seen happen in computing, so the right new idea is pursued and developed and often only later do people realise, oops, there's a security or there's a dependability issue here. Present work on AI and machine learning, or the internet of things, all of those have had this sort of trajectory. Start getting really important and then belatedly realise that there are issues like the dependability issues, which are important.

And then at that point a lot harder to get in.

Yeah.

[1:01:56]
So do you think the reason that there's less focus on dependability in industry is because of commercial pressures, or do you think there's not a strong enough communication between academia and industry?

Certainly there's a lot of commercial pressures. The whole of the computing industry has developed at very high speed and the ability to be first to market is so important that it probably makes commercial sense to hurtle along and get something into the

market and then later, when you perhaps start earning money from that, indulge in the luxury of making it dependable. Used to joke that Microsoft's systems, it was only ever the third one which was any good. But by that time they would have established a position. And it's not just Microsoft. I remember seeing an EU report. It was the report of what was effectively the scientific advisory board of, I guess it was the ESPRIT project. I've not got the name right there, but that was an attempt to set the scene for the next few years so as to justify the work programme that people then developed. And that laid out a set of possible developments, scenarios, if you like, and all of those were real gee-whiz type scenarios, none of them paid any attention at all to the possibility of things going wrong. So that what we did later was develop our own version of that report by adding a few pages to each of the scenarios indicating what went wrong. And we used that in a report to Brussels. So, there were a lot of people who either thinkingly or unthinkingly did not give dependability the sort of pre-eminence that we did. But I'd been very, very influenced by the NATO conferences and that probably more than anything else. But other things in my past made me think this way.

[1:04:41]
Of course. So, you've been in academia now about 50 years almost.

Frightening isn't it?

Had you always planned to stay this long?

No. I don't… I'm not sure that I've ever planned things very much. I've taken advantage of opportunities, so to speak. I may have had sort of strategic aims, not so much for myself, but attitudes as to what is important and what should be done. And certainly I had no thought when I came to Newcastle that that was going to be my last move. There were quite a few times when things were dangled in front of me, typically to go back to the States, but almost invariably that involved a total misjudgement as to what would attract me. The idea of some prestige job being responsible for large numbers of people and so on was exactly what I didn't want.

Okay. And this is quite a broad question, how do you feel higher education has changed in the last 50 years?

Oh dear.

Maybe more specifically computing?

[pause]

If it's too big I can split it down.

When I was at English Electric, universities were places we sold computers to, rather than places we expected to get anything from, it would be fair to say. In IBM I was turned into somebody who belatedly realised he was part of a research community and was doing research, and research for its own sake, albeit motivated by views as to what research was worth doing. So I used to joke, using a sentence that John Buxton had, I think, originated about having left the ivory towers of industry for the commercial reality, the sordid commercial reality of a university computing lab. So I probably after a while started getting an exaggerated view of how important universities and university computing was compared to industry. So, I certainly found through my work, particularly with Brussels, that it was possible to work very co-operatively between industry and university, and that I thought was a very healthy development.

And that was the ESPRIT project?

The ESPRIT project.

[1:08:36]
And can you give me a brief indication of what time period that was?

That must be the early-ish or mid-1990s. I had been a member of a small group that was sent out to Japan by the Ministry of Industry to negotiate with the Japanese the possibility of Britain getting involved in their fifth generation computer project.

There was Roger Needham, myself, Ray Atkinson was the senior civil servant, and I think, I believe, Mike Rogers from Bristol. I think I've got the names right. It was a very small group of us. The Minister had visited Japan a little while earlier and had been entranced by the offer from the Japanese that Britain should get involved in their project. We went out there and soon gained a very sceptical view of the actual research project and a very, shall I say, highly impressed view of their industry and their industry's knowledge of and attention to research, including university research. And it became clear that if Britain collaborated with the Japanese government on the fifth generation project, it would be greatly to the advantage of the Japanese industry, not to the British industry, and in saying that, that was no criticism of the Japanese industry, instead it was very much a criticism of British industry and the whole way things were done in Britain.

You thought Britain wasn't prepared to learn?

It wasn't prepared to pay attention, right? We found Japanese companies who knew far more what was going on at Manchester University than most relevant people in Britain. So, when we came back, we found that we were part of a move for a big national computer project, and that actually happened and that led to collaborations between universities and industry, but just within Britain. And certainly we had a big project here. And that was the start of things improving, so to speak. However, that whole project was coming to an end, it was clear to me that if there was going to be a successor, there was going to be a big gap, that was going to be a big financial problem to universities. I was on the UGC mathematical sciences committee at the time and so was involved in, but certainly not in any way responsible for, the fact that there was a big upsurge in funding for computing at universities. And at Newcastle that led to us having several more professors. At about that time I'd been involved in a conference session in Edinburgh…

What would you say this time was?

I'm guessing it's '93/94, but I'm not quite sure. And I was on a panel session with somebody senior from Brussels, at which I said that we as a matter of policy at Newcastle had had nothing to do with ESPRIT 1 because we couldn't cope with, we

didn't have the resources to cope with both the Science Research Council and the European bureaucracies. Then suddenly we had all of this extra resource at a time when the British government funding was going to disappear. So, that was when we made our switch to trying very hard to get European money and I got involved in this successful campaign to get a European basic research programme going. And so from then on just about all of the research that I was involved in tended to be with quite a bit of industry, even if the slight majority may have been industrial. And so all of that struck me as being extremely healthy.

Shall we take another quick water break?

Yeah.

[break in recording]

[1:14:08]
So you feel at that time, during the time of the ESPRIT project, there was a sort of a good working relationship, then, between academia and industry?

Certainly in our case.

Okay.

I'd been fairly sceptical about ESPRIT 1, the scepticism that I voiced in Brussels, that the work there had been largely industrial, that few universities had been involved, and in general that was not the leading ones. But things changed considerably. I think a fair part because of the creation of the ESPRIT Basic Research Project, which I, yeah, I guess I'm pretty proud of having had an involvement in that. And certainly I know several people who independently have talked about the impact of that. As so often happens, bureaucracies tend to get bigger and worse and a number of our aims in ESPRIT I think got enlarged and diluted, just as I think other aspects of universities and the EPSRC have. Now, it's a bit easy to look back to a mythical golden time, but I certainly remember the university as being less doctrinaire, less bureaucratic than it is now. I think of the government effect on it, the box ticking mentality and so on is

partly to do with that. I took on a sabbatical at Toronto University, in the 1970s, I suppose. Toronto University was already as big then as Newcastle is now, and when I went to Toronto I saw characteristics of that university that I find myself here at Newcastle now. And perhaps it's inescapable that bureaucracies, the task is to look after themselves, to grow and so on. So, just as I think a lot of things have improved, I think quite a few things are not as good now.

[1:17:16]
Do you think higher education is adequately preparing students to go and work in industry?

I like to think it is. I assume that higher education has got a whole range of merits and to judge it as a whole, I don't feel that I've got any particularly good viewpoint. I think that Newcastle in general does a good job. I think there are some aspects of what we teach or what we don't teach which perhaps is not quite so much to my liking. But my presumption is that we have enough in the way of contacts with industry and we've got enough in the way of reputation that we're well respected and we're well respected for good reason.

What advice would you give to someone about to start a career in computing or IT more broadly? Say, a recent Newcastle graduate.

That's difficult, because I realise that quite a number of times I've been lucky, because I've got into something at the start, and that's much easier, that there isn't so much to have to understand and to learn and so on. You can just go in and make some first draft attempts and so on, and because there isn't much else to compete with it, then you do quite well. I can't say that I had consciously planned a career like that, but that's happened several times, that the world of compilers, there wasn't much of it when Lawford and- well, first Mike Kelly and I, then Lawford and I got involved in it. The world of dependability, there wasn't much there either. There wasn't much on the history of computing and so on. So recommending to somebody now to jump into computing is jumping into a very big field and clearly within it there's all sorts of new things. And so they'll have to be jumping into a small part of something really very big, and which small part is a good one to jump into isn't obvious to me. The ones

that I know best typically are the ones that have now got rather big and being ones that I worry more about, you know, not least in things like the AI world or the IoT world and the like.

[1:21:00]
How do you see the future of computing going then? There's things that worry you?

I guess, yes. When the internet was created, that seemed absolutely brilliant and there were all sorts of wonderful things that were going to happen there. And a lot of them did. But the extent to which bad things have come as well, initially in the way of things like malware and so on, but then the appalling things that have happened with things like fake news and oh, some of the misuse of social media and so on. All of those are a considerable worry. I used to be in the habit many years ago of challenging PhD students, when they were trying to decide what to work on, with the rather sort of trite question, are you sure you will want your grandchildren to know that you were in computing. And as so often when I sound as if I'm joking, I'm being deadly serious, and the reverse. So I really meant that as a question. And I decided that one of the reasons I liked working on dependability was that I felt I could defend that from future grandchildren rather better than I could defend quite a lot of other things I might have worked on.

What do your grandchildren think?

I don't know that I've ever asked them! [laughs]

Maybe you should.

Yes. So… I can't remember what the original question was now.

I was asking just generally about the future of computing.

Oh yeah, the future. And by future- I've basically always been an optimist. In fact I find it very difficult to imagine how you can be a researcher without being an optimist. But having said that, I certainly have seen various things happen in the last

few years which have worried me. Whether one's talking about politics or the sort of things that are being done in computing and the like. And as to what might cause those sort of tendencies to be undone, it's not obvious to me.

[1:24:03]
No, I was going to ask what do you think the computing field should do as a way to combat that, do you think there's something we should be teaching students? Or early career programmers, say?

[pause] I think it might be possible to do rather more to involve computing students in activities and in volunteering and so on which is very deliberately aimed at sort of combating some of the harm, so to speak. Rightly or wrongly, I assume that in the medical school students are exposed to problems and challenged to try and do something about them.

Professional ethics?

Pardon?

Professional ethics and so on.

Yeah. And to perhaps spend time in other countries. And certainly try and contribute their skills very directly in ways that are more aimed at helping other people than at building their bank balance. Operations like VSO, is it Volunteer Service Organisation, or whatever. You don't hear much about the American Peace Corps now, but my son and now two grandsons, well in the case of my son, was involved in Project Trust and my two grandsons are now, you know, starting to try and get involved in it. That's a scheme that organises for eighteen year olds to spend a year abroad as a volunteer in a school or a hospital or whatever. And my son spent a year working in a hospital in a black homeland in South Africa, incredibly poor. And he was a handyman in the hospital, and amongst other things he built a playground there. Well, now there's two grandsons, quite independently, I don't think they even knew their uncle had done this, are involved with Project Trust in trying to organise themselves to get a year in Africa and Indian, respectively. Nothing to do with

computing there, but there's all sorts of organisations which are creatively well-meaning, so to speak, and I just wonder how much the computing world, and particularly the academic world and the academic world in Britain, does to encourage, and more than encourage, to facilitate that sort of thing. I don't know, it may be there's quite a bit. But I can imagine our turning out a lot of students that are much more thoughtful about computing and how computing can be for the general good than it actually does. I can imagine all too many students are mainly focussed on – and very sensibly – on how they can get a good job in a country where we've had ten years of austerity and so on. So.

Yeah. Let's have another quick drink pause.

[pause in recording]

[1:28:29]
Okay, so for the final topic of conversation, I'd like to ask you about your interest in the history of the computing. You've told me before how you became interested. Do you think you could tell me what do you think is the importance of the history of computing?

Well, I think there's a practical importance. Computing has developed so much on the back of technology developments that for us or for – and in particular for our students – to just come out of university with an understanding of computing as it is, is totally inadequate. I'm not suggesting that they can foretell the future, but they can prepare themselves for a variety of possible futures, or just for the variety, so to speak. And they'll be motivated to do so if they realise how different the present is from the fairly recent past, leave alone the distant past. It's almost like the difference between the information you can get from a photograph and from a movie. A photograph shows you an instant in time, a movie gives you an impression of progress, positive or negative progress. And so I think that's one crucial reason for exposing students to the history of computing. I also think that a profession that doesn't honour its sources is a pretty poor profession, that it's appropriate for somebody to know, okay, so who are the heroes, who should I be grateful to, who should I think has had a life that I should know a bit more about. So that's another fairly practical reason for regarding

history as important. In the lectures that I've given to first year on the history of computing my first lecture started with why history of computing, and those are a couple of the reasons. Another one is it's one of the subjects pretty good at getting you ready for pub quizzes. Yet another one is it's fun, it's interesting, and I think that's more than enough reasons to read and to study history.

I entirely agree. So, one of your more noteworthy achievements in history of computing is breaking the story of the codebreaking work at Bletchley Park, and of course Alan Turing is a very big name at the moment. What do you think about the way that we tell stories in the history of computing, like Alan Turing's story? Do you feel they're being told in the right way?

[1:32:25]
Fairly clearly, when things get popularised they- accuracy tends to fly out the window and *The Imitation Game* is an example of what as a film is I think a really quite good film, as entertainment. But as history it's worryingly bad. So that seems to be inescapable. Almost any historical incident that you think you know about, if you go and dig, you find it's a bit different and it's more complicated than that, often more interesting than that. Which is another reason why history is interesting, trying to find out what really happened. So, there are, on the other hand, quite a lot of very good people who are involved in Britain and America in popularising history. In computing I think one of the best things that's been done in recent years is the great book by, on Babbage and Lovelace, the comic book by…

Sydney Padua?

Yeah, Sydney Padua. I think that's fabulous.

You have a little clip from it on your office door.

Exactly. And we're going to be getting an animation from her of the Babbage machinery to display at some stage.

Oh, I didn't know that, that's exciting.

And I'm about to receive some 3D printed Babbage wheels. I thought they'd arrived yesterday, but it was something else. So, but documentaries, particularly BBC documentaries, are normally pretty good. Avoiding some of the more obvious or more appalling over-simplifications and the like. So, in that sense I tend to think we treat history pretty well in Britain. Some other countries probably might think we overdo it. I certainly think now that there really are quite a lot of people concerned with aspects of the history of computing. Though for my money I'd like the swing away from internalist histories to be reversed a bit. I think that there are too many histories which are sociological histories of computing rather than histories of computing itself. Now there's a big place for them, but I think the pendulum has swung too far. And part of that pendulum, and rightly so, is to do with issues of the disregard for the work of the considerable number of women who've contributed. But I sometimes think that it's almost impossible to write a new book on history unless it's somehow to do with sociology or feminism or whatever.

Yeah, Tom Haigh writes about that and he distinguishes, like you say, there's not so much which is history of computer science and one of the reasons that he suggests is that it requires people with both the technical expertise and also some historical knowledge as well and that such people are often quite rare to find.

[1:37:54]
A story I've told several times, but I think is worth repeating at this moment, is what happened to me when I visited the Institute for the History and Philosophy of Science in Toronto, a very impressive institute. This was in the 1970s, when I'd already done my book and I was being invited to give a talk there.

Your book on The Origins of Digital Computers.

That's right. And talking to the director beforehand, he rightly sensed that I was feeling rather nervous, a mere computer scientist talking about the history of computing, and he made a comment to me that I've treasured ever since. Which was to say, Brian, you must understand, there is as much bad history of science written by historians who don't understand science as by scientists who don't understand history.

Absolutely.

And I thought that nailed it. [laughs] So when I made my comment about internalist history, I guess there were two aspects there. When I was writing things down and sort of trying to write about the history of computing, I was in danger of concentrating solely on the history and missing out what else was happening in the world around. And in that sense it was internalist and exponentially bad. But one of the initial danger signs which alerted me to my inadequacies was when my wife realised that I assumed that Pascal was famous for his adding machine, and she pointed out rather gently the many more important things that he was much more famous for. And then something like writing the history of the Colossus. There was no way that was going to be internalist in that way because that was so much bound up with what was happening in the war and so on. So there's that aspect of what's often criticised as internalist history. And I can well understand that there's criticism of people who just write about the history of computing developments with no regard for what else was going on around them. But when it gets - I'm not sure what the opposite of internalist is, let's call it externalist – where almost all of the concentration is on things other than the computers, then I don't like that either.

It becomes something for a completely different audience.

Yes, that's right.

[1:40:05]
Okay, so I think to close, I will ask my final question. Can you tell me your favourite anecdote about Edsger Dijkstra?

[pause] I've got so many memories of him. And I've got so many memories of my interactions with him. Oh… [pause] I keep remembering my comment to him once, 'Edsger, if you ever did buy a television, I'm sure it would be black and white'. And he laughed at that, so…

I think one of my favourites is in one of his trip reports he writes about how he came to a seminar, one of your seminars here in Newcastle, on the formal aspects of computer science, and he notes that his sweater was identified as one of the informal aspects of computer science. That would have been in the late 1970s.

I think a comment made by Jim Horning was a very good one, which was that he said he'd noticed that different computing groups tended to have quite differing opinions of Dijkstra. There were some where he was regarded very highly and others where he weren't… he wasn't. And he said that he thought there was a correlation as to whether the, these feelings arose from a visit to them by Dijkstra, accompanied or not accompanied by his wife. He felt that he was a calmer, less sharp individual when his absolutely lovely wife was there to sort of keep him under control, so to speak. Toronto, he had a tremendously high opinion. It had a high opinion of him, but there were other places where that didn't seem to happen. Now, I've not heard anybody else say that, but it's a comment to make about him. And I've got very strong memories also of, not the first meeting of him, which was when he was lecturing at Bristol… at Brighton, but when Lawford Russell and I spent a week with him in Amsterdam. I remember the joy with which he was demonstrating his toaster, which was a sort of semi-automatic one, and how he had developed a really careful strategy for producing perfect toast, and so on. He was quite gadget-oriented, which doesn't fit with the memories of him at all.

That would have been when you were visiting to learn about his ALGOL 60 compiler?

That's right. He stayed with us several times and our daughter, when doing electrical engineering in Edinburgh, found people talking about Dijkstra to her. He was somebody who came and would lie on the floor and would drink milk and beer together and so on, so she had all of these funny sorts of stories about him.

Okay, well I think we should wrap up there, we've been talking for plenty of time. Brian, thank you very much.

[1:44:34]
I think I'd like to add just one thing.

Go on.

And I think it's reiterating something I said earlier.

Of course.

But the point is, I can identify just about every sequence of what I've been involved in, with a few exceptions, as when I was working with *X*, where *X* was some great individual or other. You know, starting with Mike Kelly, you know, working on through. I guess if I've got any talent, it's choosing who to work with.

Okay, great, thank you very much.

[1:45:21 track 1 ends]

[ADDENDUM]
[0:00:00]
In probably about 1981 we had a research, we were working on a research project where our aim was to find ways of producing highly dependable distributed computing systems. And our notion was to choose the most useful distributed computing system and find out how to make it more dependable. And there were a few Unix based systems around then or being planned. But we weren't very pleased with any of those and essentially, almost accidentally, we built our own. We did this – by 'we', I particularly mean Lindsay Marshall, but there were other people involved: Robert Stroud and Dave Brownbridge are other names that come to mind. We started trying to look… I'm not sure that we set out to design a distributed system or whether we had an idea, but the idea was basically, could we make a system which incorporated a number of Unix system but which overall just looked like Unix. So it was the notion almost of recursion applied to Unix as a whole. Now, I have been very keen on the merits of recursion, recursive structuring, probably ever since I got involved with ALGOL. The important differences, as far as I was concerned, between Fortran and ALGOL was the recursive, the general nature of the structure of ALGOL. So I thought of recursion, not as an addition, but as a lack of restriction.

And that had fed into quite a bit of my work thereafter. And when we were thinking about Unix, there was the idea of could we take several Unix systems and add software to them so that collectively they produced something that could be used as though it was essentially a single Unix system. And we had the idea of intercepting system calls so that we invented the idea of a layer between the application programs and the operating system nucleus so that you might have application programs that were running independently on a number of different Unix systems, but in fact communicating with each other unknowingly. [0:03:39] The Newcastle Connection was involved, therefore, producing a piece of software which was a transparent layer. It sat on top of the system calls and it provided system calls. Exactly the same system calls and given the notion of recursion, all of the system calls, every last one of them. We wanted to make sure that it could pass through our software and if necessary involve a diversion to another machine. And we realised we were thinking of distribution here as a very pure problem that we would tackle entirely on its own. We wouldn't concern ourselves with any of the other issues that a distributed system would typically be involved with, things to do with dependability or with load sharing or whatever. And so the Newcastle Connection was just that, but we then realised that that sort of notion of a transparent layer was something that we could apply repeatedly. And the simplest example of a further transparent layer was the one that produced triple modular redundancy that would allow you to run the same application program on three different Unix machines and have those machines, unknown to the application program, be comparing their results, and voting and outvoting any one machine which was going wrong and producing the wrong answers. We added that triple modular redundancy layer, if I remember rightly, at a cost of 600 instructions. We had a similar sort of layer for automatic load balancing, but then best of all, we had a way of applying recursion in the opposite direction. We'd been talking about recursion as a way of joining a set of Unix systems together to make a bigger one, then we had the idea of can we split a Unix system apart into a set of smaller Unix systems, but each one of which now is operating very securely in its own security domain, confidential restricted, ultra-secret and whatever. And that involved essentially applying the controls just on the communications between these machines. The machines using the Newcastle Connection looked like an ordinary Unix system. But we now had a set of secure communications which acted as barriers to make sure that, for example, information which was on the merely restricted machine didn't leak

out on to the top secret machine. There was a bit more to it than that, as you might say, but we went from the idea, that idea to a working prototype in under a week. The working prototype was as slow as molasses, because the encryption was done using shell script, rather than, you know, posh hardware. But we got the attention of the authorities very fast indeed on that. And that led to a secret project, which toiled away for several years, developing those ideas. Now, that I think was probably the single most protracted excited period of my research, because it pulled together both an exploitation of some very simple, very general ideas and some very practical and very clever hard work by people like Lindsay Marshall to produce something which really was quite something, something we were very proud of and that we really enjoyed doing.

[0:08:28 Addendum ends]