Capturing the Past, Inspiring the Future

# Professor Cliff Jones

Interviewed by

## Troy Astarte

18th December 2018

At the

## Urban Sciences Building, Newcastle University,

*Hello, this is Troy Astarte recording an interview on behalf of the Archives of Information Technology. The date is Tuesday, 18th December 2018 and we're in the Urban Sciences Building at Newcastle University. My interviewee is Emeritus Professor of Computing Science, Cliff Jones. Cliff began working in the computing industry immediately after leaving school with a period at LEO, the Lyons Electronic Office, before two years at IBM, starting in 1961. Cliff left IBM in 1963 to work for a year each at, first Esso doing operations research, and then Ford where he began his work in programming and development of compilers. Unusually, Cliff then moved back to IBM in 1965, spending time at Hursley in the UK and Vienna in Austria, before finishing in La Hulpe in Belgium. During this time Cliff worked on programming languages and formalism in describing languages and programs. Cliff left industry in 1979 to return to education, completing a DPhil at Oxford University under Turing Award winner Tony Hoare on a technique for reasoning about concurrency formally. Following this, Cliff took a chair in 1981 at Manchester University and continued work on formal aspects of computing until 1996. Another brief spell in industry at the small software house, Harlequin, followed. Cliff came to a professorship at Newcastle University in 1999 where he has remained until his retirement this summer.*

*Thank you for joining me Cliff, good afternoon.*

Good afternoon. It seems to me the interview's finished.

*Was everything I said during that introduction accurate?*

Yeah, that was fine.

[0:01:36]
*Great. Okay, well let's begin by talking about your early life. So, if you could tell me about your experiences at primary or secondary school, what subjects you enjoyed learning and what kind of direction you felt that would lead you in?*

So I remember almost nothing about primary school, a funny school in Hornchurch called Ayloff. I then went to a grammar school quite some miles from home, about a

dozen miles from home, called Grays Palmer's Boys' School. And there I was actually pretty lazy. I didn't really engage wholeheartedly with school, except in mathematics and certainly one of the influences on my life was a maths teacher, not the first one I had, but it must have been in about the third year at grammar school. Always called Galley , his name was Mr Gallimore , and he introduced us to the joys of Euclid. And with hindsight, this probably had a very profound impact on me. The idea that you could actually construct a proof of mathematical statements fascinated me, and in a way, I've been doing proofs for most of my career. The process actually fitted my laziness, because Galley would come into the class and say, we're going to look at the next theorem in Euclid, and put the problem on the board to prove the statement, and I was always one of the people in the class who could follow the, make the steps to construct the proof. And the homework was to learn the proof, so I had nothing to do. So unlike other subjects, like chemistry, I prospered in mathematics and absolutely nothing else.

[0:04:00]
*Okay, so you said you went to, your school was a grammar school. Was that common in your family? What was your parents' academic background?*

Oh, absolutely not. So, I don't actually know, but I would guess dad went out of school at sixteen or something and mum almost certainly stopped schooling, I don't know the age. There was no academic tradition in my family at all, so when I did decide to drop out of grammar school, this was greatly to my dad's disappointment, because he would have liked me to become the first member of the family who went off to university, and he did, just about, yeah, he did live to see me go to university.

*Can I go back just a little bit and ask you to give your date of birth?*

First of June 1944.

*Okay.*

So they were dropping V-bombs on the country and apparently we were evacuated to Harrogate, which my mother hated. And I of course remember nothing about it.

*Asides from a brief spell in Harrogate, where was it you grew up?*

Oh, in Essex. Sort of between Hornchurch and Upminster. At the end of the District Line, which will come up perhaps when we talk about first jobs I did.

*Okay. And your parents you said weren't academic, what were their professions?*

Dad was a tailor, a real tailor, I remember him cutting out suits on the dining room table. He eventually ended up in management, which was not his forte and the end of his career was much less happy, and mum was basically a housewife.

[0:06:04]
*Now, your secondary school I believe was rather strict in certain ways.*

Yes, there was a boys' school, which was the original school, and a girls' school, and the twain shall not meet. Very strict rules about that.

*I believe you have a story about that that you could tell, perhaps.*

Oh, there was a rule that you didn't, that you weren't seen with a girl from the girls' school at all, and one of the boys was seen walking down the high street with a girl from the girls' school, for which he was caned. It was his sister, but that had no influence on the punishment.

*Thanks. I was hoping that that story would come up. One thing that you mentioned is that your school had a reaction to the Campaign for Nuclear Disarmament.*

Yes, so I got interested in all sorts of things while I was at school, other than the academic things. Quite shortly after I went to school I decided I'd become a vegetarian, possibly helped by the appalling school lunches, which gave me a violent stomach problem very early in my school career. But I also got interested in politics of various sorts and one of the things was the Campaign for Nuclear Disarmament, not immediately when I went there but a bit later on. And naturally I started wearing

a CND badge at school, which the headmaster did not approve of, and when I pointed out that other children were allowed to wear Christian crosses on their coats, it didn't actually lead to a very constructive discussion.

*That's interesting. So becoming vegetarian and getting interested in the Campaign for Nuclear Disarmament, was that a response to liberal attitudes of your parents or something you picked up from peers?*

No, I was possibly partly influenced by my elder brother. So Brian was eight years older than me and naturally was more likely to be involved in things political. Although he never actually got involved in the CND movement and he was never a vegetarian, but somehow or another I was in an environment where such political things were going on around me. Not my parents, who were quite conservative.

[0:08:55]
*So, you left school quite early, didn't go on into higher education, can you explain why you chose to make that decision?*

Yes. So, Mike Waite was a person at school who was a year or maybe two years ahead of me, and he had gone off into computing, he'd gone to work for LEO computers, and I stayed in touch with him and he told me about computing. And this sounded much too much fun to sit around finishing A levels and then not being able to go into a degree in computing at all, because of course there were none in the early sixties. So I decided I would apply for a job with LEO computers. They had a very weird interviewing structure. It was a one-day interview where in the morning you were taught the straight line instructions of the LEO II order code and given a test at the end of the morning, which they marked over lunch. If you passed that test then you learnt about the branching instructions of the LEO II order code, which is of course an EDSAC derivative, you learnt about the rest of the instructions of the LEO II in the afternoon and did another test. If you passed both of those tests you got called back later to meet a human being, what would today be called an HR person. But the initial test was clearly purely technical, the initial evaluation was purely technical.

*What was it about the idea of computing that enticed you so?*

That's a hard question to answer at this distance. Remember they were very new. When actually I worked for LEO itself at Hartree House in London for a while then my joke is that I was sold with LEO 2/9 to Ilford Photographic, so I went to work for the Ilford Photographic Company who had just bought a large LEO II. When we were there, one of the things we had to do was bring people round from other parts of the shop floor at Ilford to convince them that this monster wasn't going to take over all their jobs. That's not really answering your question. I suppose I saw in some dim way that computers were really going to change things, obviously I had no idea how much or how things would develop.

*So was the LEO test your first experience with computers at all?*

Oh yes, yes. I mean we didn't have home desktop computers when I were a lad. I'd probably used a calculator, but mainly used a slide rule.

*So did you find it difficult when you encountered this new way of thinking, or did it seem to flow quite naturally from what you already knew?*

Oh, very much. I felt as though I was suddenly swimming in something that was fascinating. I should emphasise we weren't proving programs correct back then, I was just hacking out code. But it was fun, it was interesting, unlike secondary school where I – and it's not modesty, I really was lazy at secondary school – I suddenly wanted to work all the hours I could because the subject was so interesting.

*Okay. Now you mentioned that your father especially was a bit disappointed you didn't continue with education. After you'd spent a bit of time working, did your family's views change at all or did they still feel you ought to have stayed in education?*

Dad worked very, very hard and I think when he saw I was working very hard and making progress, then he probably forgot about the education issue, I think. He just

saw that I was doing something that satisfied me, that interested me, and at that stage of his life he was still doing something that satisfied him and interested him.

[0:13:39]
*Okay. So is there anything else that stands out from your early period at LEO before you moved to IBM?*

I suppose writing programs for very small machines, and I can use that to bridge to IBM. So there was a data centre in Newman Street in London, near Soho, which was conveniently placed for a vegetarian restaurant in Soho. We worked at the data centre and one of the programs that I wrote there was for a 1401. We had two machines, a 1401 and a 7090. But one of the programs I wrote was for a 1401 which had, I think, 16,000 characters of storage, but it was a linear programming program, it wasn't really a system. And that was the transition from IBM to Esso. So I got very interested in operations research. I went to night school classes where I learnt about the theory of operations research, the younger Vajda – I'll spell that properly later – the younger Vajda was teaching us. And one of IBM's major customers on the 7090 was Esso Petroleum and they were doing refinery planning using this linear programming system. And I was talking to the people from Esso and it was clear that that was an opportunity to make a jump, to move to a company which was really doing operations research very seriously, and I left IBM and the point about the strangeness in coming back to IBM, at that time IBM almost did a "bell, book and candle" ceremony if you left, they precisely did not want to pay to train staff and then have them join customers. So when I announced that I was going to resign they said, well you realise if you resign you can never work for IBM again. And being an impatient young man I left anyway and went to work with a group led by Ken Palmer at Esso, who had their offices on Victoria Street at that time. And we were using both the IBM 7090 and a 7090 or 94 at CEBG. And my boss, Ken Palmer, was a fascinating character. We had to walk across between our offices and CEGB to use this machine, and there was debate about which was the best route to get round the various buildings in the way. Ken paced it out and worked out the shortest route was through the Army & Navy store, so from then on we would walk that route. [laughs]

[0:17:05]
*So, you went from LEO, a relatively small organisation, to IBM, a huge organisation. And then to Esso, which was another big organisation.*

Which was bigger.

*But the computing aspects of it were smaller, presumably?*

Yes, that's true.

*So what were the different organisation environments like? What was it like making the shift from LEO to IBM and then away from IBM again?*

So moving to IBM was very interesting. It was clearly a very rich organisation. It was also strangely run by, now I think the managing director at the time was called Hudson, but I may be wrong on the name. On a 7090 there's a big console which has all the information about the data channels, and it has lots of flashing lights. So the managing director of IBM UK turned up and said, put that in the window, in the plate glass windows. And we said, but we need it to operate the machine. And he just said, put it in the window. And from then on we had to learn how to operate this console upside down, which was not an asset. And then another story about that time, he was coming on another visit, which we dreaded, and I was growing a beard and in consequence I was told to just get lost for the day, because they couldn't let the boss see that somebody who was dealing with customers was growing a beard. We were all wearing – all the men – were wearing three-piece suits at the time, because we were customer contacts.

*So you had to have a very smart aesthetic then?*

Yeah, which if you're operating a machine in the wee small hours of the morning is not what you want. [laughs]

*Okay. So, what would you say your early experiences working for computing companies taught you about programming? Because this is where you learnt programming rather than in a more academic setting.*

Oh, absolutely. That's brought up another name that I'd forgotten. So, remember you didn't have a laptop in front of you, you didn't have your own machine, nor did you even have a timesharing console, you had to book time to go and test your program, my linear programming system on the 1401, or a program on the 7090. In fact, there's a funny story there, which I'll come back to. But because of this booking time and getting a run and usually a core dump at the end of the run if anything went wrong with your program, you learned to be very meticulous, very careful. And I remember a programmer at the IBM Data Centre called Bruce Wall, and he would get the core dump of a run and he would explain every character in the core dump because he felt he could learn something from that that didn't necessarily pertain to the error that was manifesting itself, but he would fix other problems. And that planning and care I think is something we've lost today, because machine time is too cheap, we can just go and do another run and not think about what was really going on. So that influenced me. Do we want to transition to Ford? Because there's an operations research connection there which is funny if you…

*Sure. You said there was another story about the 1401.*

[0:21:16]
Ah, no, that was 7090. So at the time – I guess I can tell this after all these years – at the time IBM was making grants of machine time, 7090 time, to people who were considered important in the scientific world. And one of these people was Fred Hoyle, the famous astronomer. So Hoyle came along and prepared his programme in FORTRAN, punched the binary deck and ran the program, and it had a problem in it. Hoyle was very precious, very cautious of using this, I forget how many hours of machine time he'd been allocated, but it wasn't an enormous amount. So I ended up patching Fred Hoyle's binary deck by rubbing bits back into the cards in order to avoid having to do another recompilation of his program. I'm not sure this was a good use of my time. What he didn't know was that when we were running his programs, because we all knew how important it was, when we were running his

programs there was a physical clock, you clocked jobs in and out, we would clock his job out before we clocked it in so that he didn't actually record any used machine time. That's the bit I'm not sure I should talk about.

*I'm sure it's fine after all this time. Okay, so you were doing some operations research at IBM, but you saw that Esso was doing it in a way you felt was more serious, and it was interesting you at the time so you decided to move to Esso, okay.*

[0:22:56]
And that was great fun. So probably the project in my life that had most zeros after its business case was at Esso. We had a question about running crude oil through the main first unit of a refining unit called a pipe still. And if you run a heavy crude it's limited at the bottom into the pipe still and if you're on light crude it's limited at the top. So could you mix two crudes together and run them through the pipe still and get greater throughput? So this was true operations research stuff. Dick Waters , somebody who I just got a Christmas card from after all these years, Dick Waters was the mathematician, I was the programmer and we went down to the Fawley Refinery to work with the petroleum experts. And one of the first things you ask them is, we see these pipe still yields are to four decimal places, can we rely on them? And the answer came back, well, it depends whether Fred or Joe is operating the pipe still at the time, which didn't actually make the mathematical model all that much more confident. But that was one huge project at Esso. And there was another big project where essentially we had a matrix of about a thousand rows for a quarter of the year when we were planning and we wanted to work out if we could justify storing extra product between the quarters. So we essentially put these four quarterly matrices together with transfer vectors between them and showed that there was a huge saving if you had extra tankage between the quarters. That sort of project is really interesting and on the straight computing front we built a thing called … it had a name – Magic, Matrix Generator, which was a pre-system which enabled you to generate the matrices which were fed into the linear programming system. Not my linear programming system, the huge LP90 system developed by CEIR.

[0:25:35]
*Okay, so what was it then about operations research that engaged you? How did that mesh with your interest in programming and earlier your interest in proof?*

Well, I was programming. I mean the refinery people were the people who knew how to generate the matrices. They did try to get, well, I also went to night school on petroleum chemistry, but I wasn't really doing the model building, I was building the computer systems which generated the models. The only connection with proof I can give you is that understanding of the mathematics of matrix algebra and linear programming didn't interest me, that was quite involving. Let me do what I said I'd do and tee off to Ford, because that's quite a funny story.

*Of course.*

So Ford advertised for people. I can't remember whether they specifically said operations research, I think they probably did, but certainly when I was interviewed, it was on the basis that I would go and start an operations research group at Ford's Brentwood office, their head office in the UK. I interviewed and everything seemed fine, but by the time I got there there'd been a management change and the new manager – Canadian, I think he was – on my first day told me he "didn't believe in" operations research, which was a very odd claim for somebody in a huge manufacturing organisation. And I should have probably just quit there and then, but what I actually did, what I actually did was to transfer, well, to change what I was doing to become a systems programmer. So in these days every few years you got a new generation of machine and everything had to be reprogrammed. So, I was working on building the systems, generating the system for the 1410 and IBM 7010 when it came along, so really doing standard systems programming tasks. I didn't last there all that long, I'm not sure the dates are absolutely exact, I haven't got the months in anyway. I built a program there which I called a special purpose compiler. So they had a system for doing parts explosion, which means you get in the orders for the cars on, I think, three 80 column cards. It says it wants this engine and so many doors and all this, and it breaks it down, parts explosion into the bits that have to be brought to the production line. And at the point I went there, this program was an interpreter running on the 1401 or 1410, and we could extrapolate to the point where

it was going to be running twenty-five hours a day, which is not a very convenient performance characteristic. And I realised that instead of interpreting, we could actually literally compile this into machine code, and my reckoning was, it would run about five times faster. I wrote that program and it did indeed run about five times faster, everybody was delighted with it. I'm even told that that program walked when somebody left Ford and went to another company and that that program was being run at least in Ford very many years afterwards, after the machine was no longer available as such, but was being run as an simulator. And of course, one of the funny things is, because we were limited on the size of our cards, that system had the year 2000 bug in it, we had two columns for the year, and I very much doubt whether they were still running it in 1999, but if they were, that would have been the point at which they had to reprogram it. But it was a fun program to write.

*It compiled from this sort of, this system for specifying parts which was on a card and it translated it into…*

No, the cards described the car you wanted. There was a database behind it which said, if you want such-and-such an engine then you need these components to be bought. And that database, the bill of materials, it was called, actually figures in my first VDM book, because it's a really nice example of VDM data types. And of course there are some interesting data type invariants: you can't have loops, you can't have an engine needing an engine, because it would never finish expanding. So, after publishing this paper, which I got bad marks for from the Ford management: they said the only reason you can want to publish a paper is because you want to leave us. I don't think I said yes, but it was in my mind anyway. And I decided that it was probably time to do something I wanted to do rather than this system generation stuff. And I actually applied to what was then called ICL and the assumption was I was going to go there to work on systems programming, that would have been about the time they were developing their own operating system. And strangely – and I never found out what the trigger was, what the mechanism was – I got a telegram from IBM Hursley saying why don't you come and have a chat, have an interview. And of course this is against the backdrop of having been told I couldn't ever work for IBM again. And I went down to Hursley for an interview, went through a series of groups, some mildly interesting projects going on, and then I went back to personnel, what we

would today call HR. And they said, well there's one other group that wants people here, but the manager is a bit odd, Frank Williams, he's an American. So I said fine, I'll go and meet Frank Williams. And Frank was definitely rather odd, but he was fantastic, he was a phenomenal manager. And I went back to personnel and said, if I'm coming here I'm going to work for Frank Williams or nobody, he was just inspiring. [In the interview with Williams] I tried to bluff a bit, so we were talking about Lisp, and I said, actually I find Herb Simon's IPL-V language more interesting. And Frank just said, oh, write this in IPL-V. And I couldn't. But knowing you've been caught out on a bluff tells you you're talking to somebody who actually knows what they're talking about. And I went to work in the Product Test organisation which Frank ran. It meant moving to Winchester. I and my then wife moved down to Winchester. Hursley is just outside Winchester. And the first job there was testing the PL/I compiler, the PL/I F compiler. So the PL/I language was very new and the PL/I F compiler was the first IBM compiler being built for that language. And that experience of testing the compiler changed my life completely. We had a handwritten series of test cases, 635 of them, I think. And we entered what was called beta test, so these test cases were not disclosed to the developers until we entered beta test, beta test was due to be six weeks long. We entered beta test on time, but six months later we were still trying to get the compiler debugged around these test cases. When the– oh, we also designed a program which had been started by Peter Schofield, but we developed the idea a lot, called APEX, Automatic Production of Executable Test Cases, which would generate new test cases from a grammar. And my contribution there was the idea of dynamic syntax, a grammar which modifies itself with respect to the declarations so that you can generate more meaningful programs. So we could generate extra test cases. But anyway, when the compiler was eventually shipped because we had got it to run all of these 635 test cases … it was not perfect, to say the least. And I became absolutely convinced that testing was no way to get a good product out of a poor design and this was long before I heard, probably before he uttered, but long before I heard Dijkstra's famous "testing can show the presence of the bugs, not their absence". I was absolutely convinced that you had to design quality into a program. There was a senior manager in the States who actually uttered in my presence the statement, "give me a FORTRAN compiler and enough PL/I test cases and I'll give you a PL/I compiler, and it will create a PL/I compiler". And that was the absolute reverse of the way I felt about developing programs.

[0:36:42]
*So, could you just clarify for me when it was that you moved to IBM?*

IBM Hursley, I moved in 1965 and must have worked a couple of years on this test case generation and actually getting the compiler through testing. Because in April 1968 I went to IBM's Vienna Lab on a course about their Vienna Definition Language, VDL, formal definition of PL/I. So I'd heard about this formal definition and it seemed to me that if you were going to build a compiler for a language as complicated as PL/I, then starting from some formal statement of what the meaning of the language (and the syntax actually, the PL/I F compiler had a handcrafted front end, which meant that that too had many bugs in it). A funny story about that. The goto statement, the hated goto statement, in PL/I could be written G-O-T-O, followed by a label, or G-O space T-O, followed by a label. And there was another rule in the syntax of PL/I that anywhere you can put a blank, you can put a comment. So the APEX system that generated programs satisfying the syntax, eventually generated G-O, comment T-O, and this blew the compiler up and my boss, Chuck Metcalfe advised me not to walk around on dark nights for a while - the developers were not all that amused. I think they may have changed the manual saying that space wasn't a normal space, or something.

*So…*

But that was a manifestation of the syntax not being the basis of the front end of a complier, the way it would be today with a syntax generator. But I was more interested in the semantic description.

[0:38:58]
*Mm. Okay. So, just thinking about your first period at Hursley, how big was the group you were working with? Did you form good working relationships there?*

Yes, extremely good. There were a group of four of us who became very good friends, we and our wives; Geoff Crisp , John Cruttenden … The other one, I've lost his name, which is a great shame (Roger Parsons). But we would eat together and

meet together out of work with our families. I think Frank was our overall boss, I don't think we had an intermediate manager, whereas later on when I went back to Hursley between the two spells in Vienna, I worked for another important Ken, Ken Hanford – but we'll come to that later, perhaps.

*Okay. So you heard about the Vienna Lab's approach to languages and language description, and to you that seemed like much the best way to go about it, is that right?*

Well, we're getting a bit ahead of ourselves. So first of all I managed to convince my management to, Hursley management, to send me on a one-week course to Vienna to learn about the ULD, the Uniform Language Description of PL/I. This is of course all described in a paper that you and I have authored together, which is a bit difficult, but I'll try and fill in some of the details. But the personal part of it was I was there in April, the weather was delightful, I would get up before breakfast every morning and walk around the old city of Vienna and by the end of the week I was just convinced that actually, I would like to work in Vienna whatever they were doing. But also, the technical material interested me greatly. The members of the Vienna Lab had presented the formal definition along with an Englishman called George Hay , who I became very good friends with. And I went back to Hursley and said I would definitely like to go on assignment to Vienna to look at whether we could use this formal definition as the basis for a compiler design, as a basis for a systematic compiler design.

*Was that a difficult conversation to have? Did you have to work to convince them?*

Yes, I did actually. I mean whether they were reluctant to lose me or whether it just looked an expensive thing… In those days IBM assignment roles were incredibly generous, so we got, I and my wife got, all sorts of, all sorts of goodies by going on assignment for two years. Apartment found, language lessons, each of us had 100 hours' private tuition in German. But in about August that year I was back in Vienna on assignment. Oh, in between, in between there had been the IFIP Congress in Edinburgh and…

*This would be in 1968?*

In 1968. And there I had lunch at the George Hotel, which was very swanky in those days, or at least it seemed very swanky to me in those days, with the boss of the lab, Professor Zemanek, Heinz Zemanek, and the chief factotum, Norbert Teufelhart. And I went into this with a list, I think on the back of a punch card in those days, a list of all the problems that I didn't know how to solve: what to do about taxation? what to do about registration? and so on. And Zemanek would say, do you have any problems, and I'd say, well how about such-and-such? And he would just turn to Teufelhart and say "Teufelhart will solve that, Teufelhart will solve that". And sure enough, Norbert Teufelhart was absolutely fantastic. He really could solve nearly any administrative problem in the very complex Austrian environment when I got to Vienna.

*And that was characteristic of the way Zemanek ran the lab generally?*

Yes, yes. "Herr Professor". I think on the whole, no, people did use academic titles and only used family names. I remember when Viktor Kukdielka came on assignment from Vienna to Hursley, I guess, yeah, I guess while I was in Vienna, and Viktor wrote this letter back saying, you have to learn two names here. The husband's first name and the wife's first name. Whereas in Vienna of course, it was just, yeah, it was "Dr" this and "Frau" that, or maybe even "Frau Dr".

[0:44:52]
*So, how else did the cultural shift of moving to Vienna affect you? You said that you were instantly impressed by the old city, and I've been to Vienna and I can quite see why. What was it like moving from Winchester to Vienna?*

So I need to backtrack a bit. I said that my family wasn't an academic family. Also music was a very small part of our lives, although oddly, dad would occasionally walk around singing arias from operas that – mis-sing – arias from operas that I've no idea how he knew bits of. So my first wife, Jill , and I had been to some concerts. Esso had a scheme where they would get blocks of tickets to take people to concerts at the Albert Hall and so on and we had been to a few concerts. But essentially when we

arrived in Vienna we had almost no experience of really good classical music, and that completely blew my mind and transformed my love of music when I got to Vienna. By the time we left we had the tickets for the chamber music, so we would go to some of the big orchestral concerts and we would go to the opera fairly often. Of course the Viennese opera was fabulous. So, that was great.

The scientific environment was also extremely good. I immediately got into the problem that I wanted to, which was how to use a formal definition of a programming language to design a compiler. Peter Lucas, who became a good friend, and very sadly died some years ago, but Peter had already been working on his "twin machine" approach, which was an idea to be able to prove that two different algorithms for processing a concept – he would use the word a 'concept' of a language – were in step and the twin machine essentially bolted the machines together with a data type invariant, bolted the machines together, showed that they kept in step and then in his beautiful phrase, "you could rub out the machine that you didn't want and were left with the implementation technique". One of my early contributions there was to realise that actually there was a concept of the most abstract formalisation of the language and that if you did it right, if it was sufficiently abstract, then you could show by what we now call "retrieve functions", you could show by a function from the messy implementation back to the clean abstraction that the implementation algorithm did the same as the specification language. So I worked with Peter Lucas and Wolfgang Henhapl on various proofs about bits of compilers. By the time the end of the two-year assignment, a very pleasant stay in the country, great appreciation of, greater appreciation of music and enjoyment of other things in the Vienna society, I went back to Hursley convinced that this was a viable way forward. It has to be said that there were some things in those early VDL – Vienna Definition Language – descriptions of PL/I which made the job harder. Peter Lucas and I had fairly clearly identified one particular decision in the VDL definitions that made it much harder to do the proofs than it needed to be. So there was the intellectual difficulty of the real compiler design, but there were gratuitous difficulties coming from the style of definition.

[0:49:25]

*Right. So, you've written a reasonable amount about the kind of the technical work you've done in these places, so I think the amount we've been talking so far is a good level of technical depth, and I won't ask you to go any deeper into that. But, so let me ask you then, given that your assignment was always going to be temporary, how did you feel about moving back to Hursley? Did you have in your mind that you'd learnt these techniques and you were now ready to implement them, or were you feeling more that you could have profited for longer in Vienna?*

I find it a bit difficult to disentangle the two times we left Vienna. Certainly we loved the city. I think we were probably ready to go back for private reasons, family reasons, we were probably fairly happy to go back, but we retained a very strong contact with the people in Vienna, even during the gap between those two times in Vienna. I'm also pretty sure by then Ken Hanford was running the advanced technology group within Product Test, the group I'd been working in, and I met Ken and liked him very much. We'd actually written up the work on dynamic syntax together. Yeah, so Ken must have been there before I left. But Ken and I had written up the dynamic syntax work together. And I was keen to try out some of these ideas. One of the ideas – I'll resist getting too technical – but one of the ideas was of course about proving programs correct in general. And I got back to Hursley and my new colleagues, the group I now inherited from Ken, had been working on writing a version of the front end of a compiler, Earley's recogniser, and they had the program and said, "alright, if you think you can prove programs correct, here's our program, prove it correct". And I sat down and started trying to prove it correct, and I couldn't, I gave up. So I went back to a specification of Earley's algorithm and developed my own version of the implementation and then compared my developed version with their code and could spot the bugs in their code which were stopping me. And this goes back to – back and forward – it goes back to my comment about not being able to debug a compiler to make it a high quality product, and goes forward to my conviction that formal methods in an industrial environment – remember, I'm not an academic, I was not an academic at this time – in an industrial environment, formal methods pay off in the design process. Get the problems out in the design rather than produce a product with errors in it, and then use formal proof as an expensive way of debugging the program - I don't think that makes sense, and it's one of the reasons I'm less keen on model checking approaches to software than I am on using formal

methods in the design process. So that was a real example. It was a non-trivial program and it's not huge, but it was a non-trivial program and I managed to get the correct program only by starting over from the definition.

[0:53:28]
*So, it was clear to you then, that using these formal techniques, these formal methods during the design phase was definitely the way to go, but you'd said that people around you in management didn't necessarily agree with you. Can you talk about how you tried to put that message forward and why you think it wasn't received as well as you'd hoped?*

I'll tell a story that I don't think I've ever mentioned to you. There was an IBM film made, video, that was shown to all developers and it typified management's approach to this – and then I'll tell another story about it. But there was a picture, I think, of wheat fields or something with huge, giant footsteps destroying the product and nobody dared be there when the giant was there and nobody dared look at what the real problem was. And suddenly some brave person did look to see what the problem was and it wasn't a giant at all, it was a lot of little bugs eating the crop in the shape of giant's footprints. And that was pretty much management's attitude, this is just a lot of little bugs, we need to get rid of them. And again, I went to the States and met a senior manager, I was going there to make a presentation to him on why I thought formal methods were the way forward. And I walked into his office, and he knew what I was coming for, and he said "I've solved the problem". And I said, what do you mean? And he said, I've solved the quality problem. I said, oh, well, how? And he said, we're going to have a database which tracks the number of lines of code that people write and the number of APARs, which are customer error reports, the number of APARs against them, and we'll make their salary depend on the ratio of lines of code to the number of errors. Now, there's a very obvious flaw in this argument. Rather than honing down your code to make it small and understandable for the future, there was an incentive to bloat your code and have lots of code, even unrolling loops and so on, in the observation that that gave you a better ratio against the bugs that came in. This of course only became clear later on. And actually they had to set up a sub-committee to figure out how to count lines of code, because do you count comment lines? do you count blank lines? do you count a compound statement that

spreads over several lines as a single line? or what. I mean it was just crazy. But, in general, management were not software developers by background. Most of them, senior managers were mostly through sales and they understood economic incentives not technical arguments. I may be a bit out on timing, but at one stage IBM's chief scientist, Branscomb, listed the ten major technical problems faced by the company, none of them were software problems. And that tells a story about the future of IBM and there are other stories later on when I get to La Hulpe. We did some other nice things in Hursley. We wrote a new formal definition of ALGOL 60. Sorry, ALGOL 60 is a much simpler programming language than PL/I and it had already been used by Peter Lauer as a way of showing that the size of the Vienna PL/I definition didn't just come from the definition style, it came from the language PL/I. An ALGOL 60 definition is much shorter. And we wrote another definition in a style I called functional semantics in Hursley.

*I should just say, Peter Lauer was a member of the Vienna Laboratory.*

Yes, Peter was a member of the Vienna Laboratory who then went to work with Tony Hoare and do a PhD when Tony was in Belfast. And also the first ideas of VDM's program development methods, so data abstraction, reasoning about post conditions of two states and non-determinism were developed in that spell in Hursley. But if I can then move forward to the return to Vienna.

*Well, just before that, you kept contact with Vienna during your time in Hursley.*

Yes.

[0:59:08]
*Can you characterise that contact?*

This was mainly letters. So Peter Lucas, Hans Bekič and I were exchanging letters, of which I can't find very many. You know I'm an enormous collector of old paper, but I think I've only found one of the letters, maybe two. Where we were discussing what changes we would like to make in the definition style to overcome what I called the "gratuitous problems" in doing compiler development. And we probably also took a

vacation back in Austria at least, and passed through Vienna, during the couple of years I was back in Hursley. So we had developed very good contacts, particularly with the family Bekič – Hans Bekič who incredibly sadly died in a walking accident in the mountains in about ’81, ’82. Hans and his wife Sophie had seven children and of course there were some about the same ages of ours – sorry, we’d had children when we went back to the UK. And that was probably another reason for contact with them. But one day towards the end of– when must we be? ’68, towards the end of 1968, I was at home, I don’t know why, maybe it was a weekend. Peter Lucas phoned me up and said, “the Vienna Lab has been invited to build a PL/I compiler for a completely new machine range that IBM is going to be working on, called Future Systems. We’ve been invited to build the PL/I compiler and we can use whatever methods we were, we wish to use”. And my recollection is I said “I’m coming”, before he even asked if I would like to come.

*Would that not be more like ’73? ’68 was when you first went to Vienna, right?*

You’re absolutely right. So this must have been, in fact it must have been the end of, must have been the end of ’72. Yes, because there was a slight complication with my saying “Yes, I’ll come”. My son Peter was about to be born and he was born on January 14th and I wanted to get to Vienna as early as I could in 1973. We ended up with a compromise. I stayed until he was born, when he was only a few weeks old I went off to Vienna, chose a flat, which was extremely ill-chosen for a family, chose a flat and Jill came over with Peter once she’d got the hang of looking after him. The flat was beautiful, but totally impractical.

[1:02:50]
*Okay, so you were keen to move back to Vienna. Was that because of the work opportunity or because of working with that same team again, or just the appeal of Vienna?*

Or the music. I think a mixture of all of them. The people were a huge pull. I mean I make a point of saying it was Peter Lucas who called me. Hans Bekič had actually been in the UK one of the two years I was in Vienna before, so I was keen to work more with him. But just about everybody I knew in Vienna were great. Kurt Walk,

who became my manager, was a terrific individual. You've actually met him, of course. Was a terrific individual, a manager who one could really respect. In fact, something I wanted to say somewhere along the line is I really feel I've been incredibly lucky in so many ways. Remember at this stage of my career I had no university qualification whatsoever and yet here I am at that stage of my career able to go off and work on what I want to. And just about every manager in my working life, whether in industry or in academia, have been very… there have been people whom I could respect, in different ways at different times, but they were terrific. And the cultural things in Vienna. I actually took what was called a permanent transfer, so the first time I went there the conditions were so good because it was a temporary assignment for two years and you get all these bonuses, I actually moved on to the IBM Austria payroll when I went back, because I expected to stay there for the rest of my career. And we'll come later to why that didn't work out, but… So I turned up there fairly early in 1973. We immediately started writing a new definition of PL/I. Fortunately we were working with the ECMA/ANSI subset of PL/I which simplified things a bit. And travelling to and fro to Poughkeepsie, which was where the machine architecture was being designed and redesigned and changed, and redesigned, I remember spending one one-month spell in Poughkeepsie and in the middle of it, driving all the way up to Boston just to get away from Poughkeepsie for the weekend. [laughs] It really isn't a great place. It's a big factory, but it does have a development lab there as well.

[1:05:55]
*So how had the laboratory in Vienna changed whilst you'd been away?*

A bit, but not as dramatically as it changed after the death of FS. So, they had done some slightly uninteresting work while I was away, so in fact it was, you asked whether I regretted leaving, it was quite good I did leave. They'd been looking for inherent parallelism hidden in FORTRAN programs and that I think had not been a very fulfilling project. So they were very keen to grab at this PL/I compiler for a new machine. Zemanek was still the boss, he became an IBM Fellow a bit later. Oh, we'd acquired the third floor of Parkring 10, so the laboratory was not in IBM House, which was where the sales people were, it was on the Vienna ring road, the inner ring, by Parkring in an old house, which actually was above the Russian Intourist office.

And as well as the fifth floor, which was where the lab started, we acquired the third floor at some stage. But there weren't enormous changes. The changes started coming about then, because we then did acquire a significant number of new people. There was, well, there was a real American, Bill Schmidt came in with his family and eight hunting rifles. There was a pseudo-American, Franz Mayrhofer, who from the name you will guess was actually a born Viennese, but he came in on American assignment roles. And Dines Bjørner was hired actually from IBM in the States. So… And there was another American who came and didn't last very long.

*Dines Bjørner, a Danish…*

Dines is Danish, his wife Kari is Norwegian, stayed dear friends ever since. But he had been working at IBM's San Jose lab, San Jose research lab. And he and I have always disagreed about where we first met, it's completely bizarre. We met in the States somewhere, but he has one version and I have another.

*Do you want to get yours down on the record?*

I'm not sure I can remember the details now, but I can remember we've often disagreed about it. So maybe I should accept his… [laughs]

[1:09:10]
*Fine. So you ended up working quite a lot with Bjørner editing a book together, multiple books actually. Was there something about your working partnership that was particularly profitable, or was it more a kind of, it just happened naturally?*

I'm not sure whether you've ever heard these stories and whether you're asking that tongue-in-cheek. Dines and I had some quite firm rows on occasions. I can remember when we were working on one book, he was doing the typesetting in Denmark – so this is jumping ahead, this is after the death of FS – he was doing the typesetting in Denmark and I was trying to get material together, and I remember standing in my kitchen on the phone intending to tell him, "I'm pulling out of this book because we can't work together any more". And he blasted down the phone about all the things he thought I was doing wrong, and we agreed I would fly to

Denmark and we'd sort it out over a dinner, which we did and we stayed good friends. But our working styles were somewhat different.

*So let's talk about something less controversial then, the cancellation of the project that you had been working on.*

[1:10:36]
Yes. So we had pretty much, well, we had the full definition of PL/I, we had a formal description of the machine, the FS machine, which was non-trivial because it was a very high level machine in itself, so we had to have a formal description of the machine we were mapping on to. And then IBM for purely commercial reasons, decided it wasn't going to go ahead with the machine and we got the message that the machine was being scrapped and that clearly left our project doomed, the project had to be cancelled. People started leaving. Actually, Peter Lucas I think might have left and gone to IBM in the States a bit before the cancellation of FS, I'm not sure about the order there. But people started leaving, Dines Bjørner left fairly quickly, Wolfgang Henhapl left to a professorship at Darmstadt. Dines went back to Technical University of Denmark. I spent a long time trying to help essentially Zemanek set up a new mission for the lab. I remember one visit with Kurt Bandat to Paris where we, (Paris was where headquarters was and we'd been talking to some commercial organisations about doing some work, obviously for IBM, but that would have helped this commercial organisation). And Kurt Bandat and I got back to La Défense, headquarters building, and said to the secretary, we need an empty office to go and talk. And she thought for a moment and gave us the keys to an office, which we went to, and it was empty, there were no chairs, no desk, nothing. I think I might have broken down crying, but with Kurt Bandat we just both burst out laughing. I mean this was just a measure of the mess we were in. Zemanek actually asked me if I would stay on in Vienna (So it quickly became apparent that, or it became apparent over a period of time that we weren't going to get any really interesting work for the Vienna laboratory) Zemanek asked me if I would stay on in Vienna as one of his *wissenschaftlicher Mitarbeiter* [scientific assistants] and I didn't see that as my future at all and I fixed to move to the IBM European Systems Research Institute in Belgium, which was essentially offering three-month courses for experienced IBM engineers and I went there to teach formal methods.

[1:13:54]
*So, before we move to Belgium, I'd like to ask a little bit more about the cancellation of the project. Not…*

[laughs] Painful memories.

*Yeah, I'm sure. So that, I think in February 1975, is that right? That's when it was cancelled?*

I think so, yeah.

*So obviously it was a big blow for you. Do you feel that had it gone ahead you might have been able to, had you been able to go ahead with creating this compiler with the formal methods built into the design process, that you would have been able to use that to really show off how useful these tools could be?*

Oh yes. I mean apart from the PL/I definition itself, if you look at the technical reports, and unfortunately even the PL/I definition was never published in the normal sense, it was only a technical report, but if you look at the other technical reports which were written with colleagues like Itzbicki and Weissenböck , you'll see that we had all the bits that we needed in order to proceed formally. It would not have been a machine checked proof in the modern sense that you would expect a compiler to be developed formally, but we would have informed the whole development process by formalism and I believe produce something really exciting. So it was a terrible blow when it was killed.

*And do you think that might have possibly been able to shift the culture amongst some of the IBM management that these methods were too costly to be worth it?*

I would hope so, but something you have heard me try to explain to some of the historians we now meet in our recent work, is that IBM never was a single culture. So

there was a part of IBM called Federal Systems Division – probably still is – FSD, where Harlan Mills was pushing a particular brand of formal methods. I didn't actually agree with everything Harlan was trying to do, but he was extremely effective with the management in FSD. Partly on the back of being able to convince the division director, so the man right at the top of FSD, that this was a good way to go. So when he started running courses on his approach to developing programs, the division director came on the very first course, which meant nobody in the division could refuse to come on the course. Again, we're getting a bit ahead of ourselves, but when I was teaching VDM courses to particularly IBM Böblingen, but within Europe, it was a much looser process going on. We could never get the managers to sign up for the courses. They had in the end had a special readers' course. I can tell a lovely story about Italy. I had a very good friend at the time, actually I worked with him in Belgium, Attilio Stajano , and he fixed up that I taught one of these VDM courses in Urbino, north-east Italy. And having had the people in Urbino for a couple of weeks, we went back to Rome to present back to their management. And his engineers, his Italian colleagues, started presenting a formal definition of something with the head of the Rome Scientific Centre in the room. And this guy, after about quarter of an hour, just got up and walked out. So Attilio followed him and I of course had my heart in my mouth because I thought the whole thing had gone pear-shaped, and apparently this senior manager said to Attilio , "I can't understand a thing, what they're writing about, what they're doing". And Attilio says, "That proves it was worthwhile. If you could understand everything they've done we wouldn't have needed a two-week course to teach it to them". But we could not get the managers in Europe to sign up to the course. So that, again, in the future, when IBM wanted to do the CICS definition, although we had had VDM courses running for some considerable time, there wasn't a cohort of people in Hursley who could write VDM specifications. I was just finishing my doctorate in Oxford, about to move to Manchester – in fact I probably didn't have the job by then – and Rod Burstall and I, who were consultants on the project, recommended the use of Z, simply because there was expertise in Z in Oxford, which was much closer to Hursley than I would be in Manchester. But, compared with what Harlan Mills managed to do where he got a sort of top-down management acceptance of everything, and going right back to your question about would it have had an influence on IBM, what one has to realise is that some things happen in one part of IBM and are very reluctantly, or very slowly, or maybe not even

at all, adopted by another part of IBM. A classic story is relational databases. Ted Codd was treated as a sort of guy out on his own for a very long while, but eventually IBM abandoned its hierarchical databases and did start using relational databases. But different bits of IBM were doing totally different things.

[1:20:29]
*Okay. So can you tell me a bit about La Hulpe, what you were doing there?*

Teaching. So we ran these three-month courses. There were some shorter courses as well, but basically I think I usually taught on the three-month courses. And essentially I developed the *Software Development: A Rigorous Approach*, 1980 book on VDM, while I was there, through the courses. So I taught the courses on the programme development parts of VDM and wrote the book in my spare time. Also, picking up the story with Dines Bjørner, we initially were trying to get a collection of papers for the *IBM Journal of Research and Development*, the more technical of the journals that IBM produced at that time, which described the work that had gone on in Vienna and, as we've said, had to be stopped. I don't actually remember quite what went wrong with that (J of R&D), but we decided, Dines and I decided in the end, we would collect the papers together and make a Springer *Lecture Notes in Computer Science*: LNCS 61. Which I think there were probably some tensions with Dines during that because I rarely did things with Dines without argument. But that was a much simpler operation and that book came out quite quickly. Yeah.

*So that was your first experience teaching then, whilst you were in Belgium, is that right?*

Pretty much, yeah. Certainly first experience teaching that sort of length of course. Obviously I'd given odd seminars and talks. I remember my worst seminar ever - I don't know when you'll have to say this in your career – but we, Peter and I had done this proof of the block concept implementation together.

*That's Peter…?*

Peter Lucas. And I was really pleased with it, so I got invited to give a seminar at Queen Mary College back in the UK and I was going to drag them through every line of this proof [laughs] and there were rather less people in the room when I finished the seminar than when I started it, and with hindsight, it was an absolutely awful way to approach a talk. But in answer to your question, yes, in La Hulpe was the first time I really taught in an organised way.

*And did the teaching, did that help you, you said it led to the publication of the book, so did it help clarify the ideas in your mind?*

Oh, absolutely. You know, having been with me some while, that I clarify ideas by trying to explain them. I think most people do actually, but some people manage to do it with less student suffering than I do. But over time I clarify my ideas by explaining them.

*Okay, so we're going to end the recording here for now and resume again after a brief break.*

[1:24:19]
*This is the interview conducted by Troy Astarte with Cliff Jones on 18th December 2018, part two.*

*Okay, so you'd spent some time at La Hulpe in Belgium, teaching courses, essentially. And then after that you moved to Oxford to complete a DPhil. What was it that caused you to take that change in direction in your life?*

So I had been interested in taking an academic job earlier, quite a lot earlier, I think. Dave Cooper, who is one of the unsung heroes of UK computing theory, of computing, had advertised some posts at Swansea and I had applied but for some reason he preferred to appoint Robin Milner (Turing Award winner), subsequently as Turing Award winner. So obviously this idea of academia had been on my mind. In Dave Cooper's defence, of course, I had no qualifications, no university qualifications at all at that time.

So I was on an IBM trip, may have even been an IFIP WG 2.3 meeting, I can't remember, but I met Tony Hoare and Tony mooted the possibility – I think he initiated it, but I don't remember – of me going to Oxford to – in Oxford one registers for a Masters degree and then transfers to a doctorate. Funnily enough, when I got back to my office in Belgium, there was a letter awaiting me from Brian Randell, which in typical Brian style said, "do you think it's time to regularise your resumé"? Meaning, get a decent qualification. But I'd already pretty much committed to go to Oxford to work with Tony. Of course there was the little problem that I didn't have a Bachelor's degree and like all good universities they have rulebooks and the rulebook in Oxford explains that you have to have a Bachelor degree before you can register for a higher degree. But, being an organisation which has been around for many centuries, if you look at the bottom of the page, it tells you where to go to get the exception approved. That is a mature organisation and more organisations should adopt that. When I applied to the Research Council for a grant to do a doctorate, remember I had had no taxpayer's money at all to get tertiary education at this point, I was refused point blank because I didn't have a Bachelor's degree. So I worked my way through college by teaching courses and consulting for IBM, mainly IBM Germany, but some other bits of IBM.

*Whilst you were doing your DPhil?*

Whilst I was doing the doctorate, yeah.

*And yet you still completed it in two years.*

Yes, yes. Well, I mean I'd already written a couple of books and quite a lot of papers. I did actually know what it took to write and of course, being able to write, just like you can, is an important step to being able to produce a PhD. I also had a pretty clear idea what I wanted to do. VDM as we had it essentially had nothing to help with concurrency. Hans Bekič had made some attempts, which he never properly published, but I knew what I wanted to do and fortunately I found a way forward, the rely/guarantee idea, relatively quickly. And it was then just a case of working it out. But yes, it was a fun two years in Oxford working with Tony.

[1:28:58]
*Did it feel completely different to working for an industrial organisation?*

Less different than you would think, because I walked through the door and Tony had just started an MSc and he would grab anybody he could to teach on the MSc, so I was there again in Oxford teaching on the MSc while trying to do my own doctorate. But they were a super bunch of people. At one point I shared an office with Jean-Raymond Abrial, who again became a very good friend and of course is the father of all sorts of specification languages: Z, B, Event-B. That was fun. At a later point I shared an office with Lockwood Morris, who sadly died a couple of years ago, and Lockwood was a great colleague. Partly because we were sharing an office, but pretty much I worked early and Lockwood worked late, so we didn't interrupt one another very much. And we wrote the paper together where we exhumed Alan Turing's first attempt to prove programs correct.

*What was your relationship like with Tony Hoare at the time? Did you work closely with him?*

No. Tony at that point was inventing, evolving CSP and the people who were working closely with him very much on his agenda were Steve Brookes and Bill Roscoe. CSP is a process algebra where you don't have any state, roughly you don't have any state, and I was working on shared variable concurrency, so we were working somewhat separately.

[1:31:01]
*Okay. Whilst, another thing at Oxford – it seems you did plenty of things at Oxford – you said you were also in Wolfson College.*

Well, I was a member of Wolfson College.

*Right.*

One has a college affiliation in Oxford, graduate students who didn't have an earlier association with an undergraduate college, so Bill would just go back to Univ where

he'd been as an undergraduate, but people like me often graduated, often ended up at one of the graduate colleges and Wolfson was a wonderful place to be.

*Right, okay. That being, that had links with the Programming Research Group right from the beginning. Did you feel there was a particularly close connection between the college and the group?*

Yeah, most lunchtimes a bunch of us would walk up from 45 Banbury Road up to Wolfson College to lunch in the college. It wasn't fantastic food, but it was quite good food, and one of the things I liked about Oxford, you never had money on you. You charged your lunch to "battels" and you went down to Blackwells and charged your books to the account and, what did you need money for.

*Food and books is all you need as a student, eh?*

[1:32:36]
*Okay. So, following your time at Oxford you went almost immediately into a chair at Manchester University.*

It was immediately actually, in fact. I was appointed to the chair of Manchester before I'd been awarded the doctorate, which was a very bold decision on their part and I thank them for it to this day. After I'd been there for a while I was on a chair committee appointing a professor of something or the other and one of the other members of the committee said, "you're the guy who we appointed with O levels". And I said, oh, it's alright now, I've got my doctorate. And he said, how boring, it was much better before. [laughs]

*So, just quickly should mention some dates. Your DPhil was '79 to '81.*

Correct.

*And so you took the chair in Manchester in '81.*

Correct.

*Okay. How do you feel your work as an academic grew from your time in industry? Do you feel there was a huge difference in the kind of the research that you were doing or the way you had to do research?*

[1:33:49]
No, I don't feel there's a huge difference. I think something about my research is it's always been driven by problems. So I cannot sit down and work on something purely for its mathematical beauty. I don't think that makes it applied, but I have to know that it's applicable, I have to know that it will solve somebody's problem. And industry was a wonderful source of problems and when I went to Manchester I consulted a great deal. In fact, we had a wonderful vice-chancellor, Mark Richmond, who went on to lead the Research Council, and then went after that to head up research at Glaxo. But Mark called me in to his office one time and essentially said you're spending enough time consulting that maybe you would like to volunteer to reduce your salary take from the university. But in those days we were allowed to do twenty per cent of our time consulting without anything, mine had just got beyond twenty per cent, and that stimulus of working with industry stayed with me even while I was in academia. And still does to a large extent.

*Do you feel that's quite unique to you?*

No. No, no. So, look at people like Tony Hoare. He built compilers and then by his own admission failed to build an operating system in industry; Rod Burstall worked in industry; Robin Milner worked in industry. I'm convinced that good formalism comes from an inspiration of problems we have faced. I don't think – you can develop beautiful mathematics – but I don't think you can develop an applicable mathematic approach without having tried to really develop something. And in many ways I regret that I don't program as much these days as I used to.

*Do you think there's enough collaboration between industry and academia generally?*

No. That's not a long answer, but it's my view.

*How do you think more could be prompted?*

This is really interesting territory. So, I think getting away from the bean counting that goes on in academia, so counting papers, which is not quite what they do, but counting citations to papers is quite close to what academia does, is never going to yield a productive environment. I know that there is now an attempt to measure impact, but it's terribly difficult to measure in any meaningful way. Yeah, I shake my head about what's going on in academia today. I think ill-informed outsiders are trying to tell academics how to be effective and academia has been effective for many, many years. It's not the right way round.

*Does that perhaps mirror your experience working for IBM on what you thought were really useful techniques and then having the management not pay attention to them because of metrics or commercial reasons?*

It's a very interesting line of argument, but no, I don't think it is the same. I think the similarity is that people in control maybe need to trust the technical people more, but in IBM, so in IBM one of the key measures for salesmen was the Hundred Per Cent Club. If you- you got given a quota at the beginning of every year, if you were a salesman and you sold a hundred per cent of your quota, you went swanning off to some nice place round the globe and got looked after. Funnily enough, they tried to introduce something like this for technical people, a reward for leading technical people, and I got shipped off to Montreal, and we had fascinating talks. The guy who did the first heart bypass operation was one of the speakers and Teller was one of the other speakers.

*Of Penn and Teller? Teller who?*

The bomb Teller [Edward Teller].

*Oh right, okay.*

Who was terrifying. This was not all that long after the Prague Spring. Well, it was a bit after Prague Spring, but essentially his argument about why we had to have yet

more and more bombs being developed was "look at the alternative - you can try negotiating with the Russians and look what happens". I just found it a completely scary experience and so did quite a lot of other people in the room. But sorry, back to your question. I think technical people are actually pretty good at motivating themselves towards applicable problems. If you start bean counting – Fred Schneider said this beautifully when he first heard about the research assessment exercises in the UK. He said – obviously I'm paraphrasing – he said, if you're going to measure something and reward people for it, you'd better make sure it's the thing you really want to optimise, because people will game play against what you're measuring. And that's happening. The length of author lists on papers is getting longer and longer, because all of the authors can claim the bean of that paper. Is more useful research being done by that process? Of course it isn't.

[1:40:58]
*So, what do you feel then are the big challenges to computing research in the future, aside from trying to sort out how to best reward the right kind of research output?*

Persuading people that quality is, persuading customers that quality is worth paying for. If the game is that every new feature that comes out has to be grabbed by everybody and has to be on everybody's iPhone or whatever – this is not an anti-Apple comment by any means – if it's always new features which drive us forward, then quality will suffer. What we want is people to say, I will pay for something which I know to be correct, something which I can rely on, something which I can sue the manufacturer if it doesn't meet its specification. Then we could bring about a change. But at the moment software is of very questionable quality, you read it in the newspaper all of the time, but the customers are still prepared to grab the latest new feature in the belief that they can't live without it.

*And what do you think the role of economic concerns is in that?*

Well, that's why I say I think something like the car industry where if a car injures or kills people, then the manufacturer is liable, we cannot have a situation where it says on the packet – sorry, we don't get our software on CDs any more – but it used to be

if you open this packet you give up all your rights. That's just unrealistic. I mean companies lose business because software has bugs in it. It's happening all the time.

[1:43:16]
*We've rather skated over your time in Manchester. Is there anything you want to talk about from that period?*

Oh, it was a great period.

*What do you feel your greatest achievements were from that time?*

Well, I got a five-year Research Council fellowship, which was brilliant and I did a piece of work there that I still feel maybe hasn't come to full fruition. A different approach to concurrency, a more proof oriented approach, transformational approach. I had some fantastic PhD students. Lynn Marshall came up with a problem that at first I was convinced she just wasn't trying hard enough to find the most abstract definition and she eventually convinced me, I couldn't find one. It was a very nice problem and led to Tobias Nipkow's PhD where Tobias figured out how to cope with the situation where there wasn't an appropriate most abstract specification. Ketil Stølen did great work extending the rely/guarantee type thinking to deal with progress arguments, which hadn't been done before. Jen Cheng did a very nice PhD on partial function logic. And towards the end of my time there, there was a Newton Institute event in Cambridge and I spent six months in Cambridge at the Newton Institute, which is a wonderful place, you must go there, and we were looking at issues around semantics and another individual, who's died, sadly, too young, Gilles Kahn. Gilles was a great force for fun at the Institute, one always knew when Gilles was in town.

So yeah, I had a lot of fun there [Manchester]. But towards the end of that time, I'd had contacts with the boss of Harlequin before, Jo Marks, and I got in discussion with Jo again and Jo asked me if I would go and head up a division in his company, which was about 300 people. So I headed up the Information Systems Division for a year and a half or two years. The company was sort of being run on a shoestring because Jo was convinced that we should build a compiler for a particular language which really didn't work out, but sunk a lot of money in it. And eventually the company got

in rather more diffi… I became technical director of the company, partly because we were eventually going to venture capital, Jo had built the company solely on bank borrowing, but he was finally going to venture capital. And I had to present the technical portfolio. Venture capitalists would come in and I would explain the technical portfolio and everything would be rosy. They then start looking at the books, the financial position of the company, and chose not to invest. And we had to go through a lay-off, which I thought was quite well managed. Jo let the division directors, so I and three or four other directors, decide who had to be laid off. And it wasn't nice, but we went through with it and the company was a bit leaner, but it was all fixed over a weekend and we told the people we had to let go on Monday morning. Then there had to be another lay-off which Jo himself chose to run and I decided to quit. When I quit, I had a drink with my colleagues on my final day at work lunch, went home and got a telephone call to say the company was in receivership, which at that point owed me quite a lot of money. [laughs] I did get it all in the end.

[1:48:05]
*So what did you learn during your period working for Harlequin?*

Oh, I learnt a lot about myself. A lot about myself. Brian Warboys at Manchester, he was a Prof appointed at Manchester, predicted that Jo and I wouldn't survive in the same company six months. In fact we survived three years. But I'm convinced that one of the reasons that worked was because I handed in my resignation on the first Friday I worked for Jo. And he knew I would walk if I didn't agree and we worked together. I didn't always agree with him, but… In fact, I often didn't agree with him. But it was a really interesting time. I could never have formed a company like Jo formed, I could never have built up a company to 300 people, I could never have made the tough decisions he made. I didn't do … I did hardly any scientific work during that three years and that had a really strong effect, and I've spoken to other people about this. John Fitzgerald, who was also a PhD student in Manchester, but John from here went off to work in industry for a couple of years and the comment I made to John is, it's not just the gap while you're out of academia, you come back into academia, both of us to Newcastle as it happened, with an empty pipeline. If you lock that door today, I'll still be producing papers or books in a year or two's time. When you come back with an empty pipeline, you've really … it takes a while to get

back up to speed again. I did. And I've enjoyed Newcastle very much. I had one full offer and one job very close to an offer for universities in the south. When I called up Brian Randell and said there's this new call coming out for Interdisciplinary Research Collaborations, could I join in one with Newcastle if I go to this other university. And Brian said "have you signed the contract?" And I said, well no, actually. So he said, come and have a chat. And I came to Newcastle because I found it was an incredibly supportive environment. At the time I came here under John Lloyd's leadership, I felt this place really had a good collaborative working environment.

[1:51:16]
*Okay. Well, let's talk a little bit more about higher education then. Specifically I'd like to ask you about teaching. So you say you've done a good amount of teaching work throughout your career. Now, in the early 1970s Christopher Strachey said he didn't want to teach an undergraduate course in computing because it would only be a state of the art mishmash and would not teach proper conceptual ideas. Do you feel that still holds true today?*

No, I don't. I feel there is a body of knowledge and in many places it's quite well-structured knowledge that we can teach our students. The problem is, it's not absolutely clear that the students understand what their best benefits are. They tend – and some of the employers tend – to think that what a student coming out of a university course should be ready with the latest toys that are actually going to be very ephemeral, that are not going to be relevant in another five years' time, certainly ten years' time. So you know that my view is what we should teach is techniques for understanding things, because one can then go on and employ those things with whatever you have to meet later on. But I do feel that body of knowledge is there and we can teach it. I'm not disagreeing with Strachey's judgement nearly 50 years ago, I'm saying that I think the situation has evolved.

*Do you think that we are now properly preparing students for a career in computing?*

I think we're doing a reasonable job. We [Newcastle] are, as you know, revising the [undergraduate] syllabus over the next three years. I mean we're revising the first

year now and we'll be changing the subsequent years in a way that I hope will make it even more relevant and more useful. I should say about teaching, since you are taking over my course, I greatly regret not teaching. I've loved teaching all the way through. There's a lot of talk about research led teaching. I feel my research is often driven by my attempt to explain things to students. It's not just exposing the students to some current research; it's the feedback from, "I didn't explain that terribly well, oh well, perhaps it's not thought through properly, perhaps I should go back and do some new work". And sometimes students ask incredibly good questions that force me to think things anew.

[1:54:21]
*So how then, having spent a lot of your career in industry and then also in academia, what advice would you give to somebody, an undergraduate leaving for a career in industry?*

[laughs] If they're research minded, I would always say that there are some advantages in being in industry. If you can shift, particularly a large industrial company, but if you can shift the direction of an industrial organisation, it has an effect fairly quickly on products. But then there's all the warnings. Projects will be killed for commercial reasons that have nothing to do with their scientific value, so I think over half of my PhDs have gone in – I must have had some twenty-odd PhDs – and over half of them have gone into industry. Some of them have gone in and come out again – Kevin Jones went to a start-up in California and is now a professor in the UK, back in the UK.

*Do you think being an effective researcher requires time in both academia and industry?*

Requires is too strong, but I think it's beneficial, for the reasons we talked about a little while ago.

[1:55:56]

*Okay, alright, I'll make this my last question then. International collaboration is a big flavour of both your industrial and academic periods, how important do you feel that international work is?*

Oh, incredibly important. I think the biggest thing that the EU science funding programmes did was to make academics, more academics, more aware of one another. So of course there were always a few people who knew academics in other countries, knew a few academics in other countries. Those big EU funded projects on the whole didn't yield huge scientific breakthroughs for the quantity of money that went into them, but they established collaborations which will exist for decades and – dare I mention Brexit at this stage – I think Britain could lose a great deal from attenuating its contacts with academic research within Europe because of Brexit. But that really ought to be a final note before I get too political.

*Okay. So I'll just give you a minute or so if there's anything else you feel that you really would like to say that you haven't yet had the chance to say. What should I have asked you?*

No, nothing you should have asked. I'd like to repeat just how lucky I feel. Dropping out of grammar school was a rash decision, maybe. I've worked with tremendous people along the way and had a career that I just couldn't have imagined. I did not, certainly did not imagine, but I could not have imagined at the start of my career. It's been fun, and continues to be fun.

*Yeah. I had a question, why choose to retire? Do you still plan to keep working? Will you ever stop?*

My joke is, I tell Joanne, my wife, well, "when I retire …". [laughs] Meaning, nothing's changed yet.

*Okay. Well, we'll end there, thank you very much, Cliff.*

Thank you.

[1:58:31]
*Okay, this is Troy Astarte interviewing Cliff Jones for the Archives of IT on 18th December 2018, addendum. So, one thing we haven't discussed is the DIRC research project, which was a large interdisciplinary project that ran for many years at Newcastle University and other places as well, can you tell me a bit about that project?*

Yes, in a sense we skirted on it. When I said that I called Brian up and said the IRCs, (Interdisciplinary Research Collaborations) were being talked about, that was the call to which a hundred and something bids were put in. So I came here, joined in the preparation of the bid and the hundred and something applications were whittled down to twelve and then we had a shoot-out in London. So twelve groups went down and presented their cases, and eventually five of them were funded. So DIRC was the Dependability Interdisciplinary Research Collaboration, D-I-R-*C*, although there was always a thing we should have a dirk, a Scottish dirk, as our symbol, and David Greathead, who was one of the people on the project, found one that I could have picked up for just a few hundred quid [laughs], and kept as a symbol of the project. Given today's knife crime, I'm not sure I'd want it around. We got funded.

*Who was the funding provider?*

The funding was the EPSRC – Engineering and Physical Research… Science Research Council. It was a lot of money, we were funded to do research for six years initially and there was a bit of an extension. Initially we were funded to the tune of about seven million pounds and we got some extra money along the way.

*When was it this started?*

So it started in, it started activity in 2001 and finished in 2008, I guess. The five universities involved were City University in London, Bev Littlewood leading it from there; York, Alan Burns; Lancaster was led by Ian Sommerville and Edinburgh was led by Stuart Anderson. And Newcastle was the overall leader of the project, so the money actually came to Newcastle and Newcastle funnelled it out to the other… It was an enormous risk and fascinating exercise. We had, of course, some computer

scientists, but we had psychologists, sociologists and statisticians working together. Now, I mentioned much earlier that I'd worked in operations research, and I kick myself that it took me a year to remember the message from operations research. So for the first year we all sat around and sociologists discussed what they meant by a method and computer scientists explained that that's not a method to them and we, alright, we were forming a coherent body of people but we were in a sense talking past one another. And then we dropped a problem on the table, which is what you do in an operations research group, and you say stop arguing about terminology, work on the problem. It happened to be reading mammograms, so a radiographer looks at an image, forms a judgement on it and passes it to another radiographer who forms an independent opinion and they then come together and decide what needs to be done. And there was an experimental computer system which would replace the second radiographer, thus it would be a radiographer and a computer system. Before anyone runs to the conclusion this is a way of saving money or whatever, it meant that more images could be processed, so it was a productive economy in the sense that it would boost the effectiveness of the programme. But in true clinical style, what happens is, you average all of the results. So if you're giving drugs to people you have to average the results to find out whether the drug is generally beneficial, and that's what had been done with the results of reading these images. And pretty much the result was the computer system was mildly useful, but only mildly, but that was the average result. What happened when the statisticians started looking at the detail, if you pull it apart, the really surprising conclusion was that the people who were less confident, maybe less accurate, but maybe less confident with reading had their performance improved by having the computer back-up system, but the most reliable readers actually produced worse results using the computer back-up system and eventually it was clear that the computer system tended to make the same judgements as a human being would, so it was just a reinforcement thing. And if you pulled the results apart you got a very different picture. The sociologists went in and looked at how radiographers actually relate to the surgeons, the computer scientists looked – there was another one of the IRCs which was heavily involved in image recognition, Mike Brady's IRC. And by dropping this problem on the table and getting all the different disciplines to look at it, we formed a much more coherent project. And it was just a fun project. We really went through some good learning experiences and I wouldn't have missed it for worlds. Of course, it was little to do with formal methods, but I

guess, although most people regard me as a one-trick pony, I'm not, I enjoy doing other things. And those collaborations have lasted, so I'm currently doing research with Alan Burns at York, on a slightly different topic, but it is something that came up during the IRC and now I am combining my formal work with Alan's work on timing. So it was a really interesting project and I don't see that sort of funding as easy to get today from the research councils, as it was in the year 2000, I guess.

[2:06:11]
*I was going to say, it feels like the very interdisciplinarity, although a challenge, was a big part of the success of that project and do you feel that enough emphasis is being put on interdisciplinary work?*

Well, the short answer is, if you talk past tense and maybe up to today, my answer would be no, I don't. I do hope that the coming of UKRI, which there has certainly been a lot of discussion – sorry, UK Research… no, I'm not sure what UKRI is [UK Research and Innovation]. I do feel there is more discussion about interdisciplinarity now. I'll tell you one funny story about the shoot-out when we went down for our funding, putting our funding case forward. The chairman of council at the time, who, chairman of the Research Council at that time, was actually, I guess, a botanist and he had his panel, big panel stretched around the room, asked us various technical questions about what we were proposing to do. We'd of course made a presentation, we were asked questions about the presentation. And at the end, the chairman of council said, well, I'd just like to ask a closing question – I'm paraphrasing – "something like this IRC could function like a eucalyptus tree to our research". At which point my mind went completely blank and I had no idea what to answer, but fortunately Robin Williams and Michael Harrison – two other members from the project, Michael at that time was at York and Robin was at Edinburgh – two other colleagues were with me and one of them quickly stepped in and said, apparently a eucalyptus tree destroys everything around it, no we wouldn't function like a eucalyptus tree. But it was one of those moments when you're just incredibly pleased that you're a group who know how to work together and toss questions to another person.

*Great. Okay, thank you very much.*

[2:08:40 recording ends]