Tony Storey

Interviewed by

Richard Sharpe

20th July 2021

Via Zoom

Welcome to the Archives of Information Technology, where we capture the past and inspire the future. It is Tuesday the 20th of July 2021, and I'm Richard Sharpe, and I've been researching, covering and writing about the IT industry since the early 1970s.

[00:20]

The glitzy part of computing and IT nowadays is graphical user interfaces, and things that help us search the Web, search engine optimisation for example. But behind that, there's a whole amount of software, which is far more reliable than your, your Microsoft or your Apple operating systems, has to be, because it is the very heart, the guts of the whole of the IT systems that we're resting upon nowadays. And it's in that area that Tony Storey, who makes his contributions to the Archives today, has mostly made his contribution. But before he did that, of course he had to be born, and he was born in 1941 in County Durham.

[01:08]

Tony, what was the background of your parents?

Well, I didn't know my… My mother was just a, you know, in those days she was just a, a housewife, looking after the house. My father unfortunately at that time was, was obviously called up into the Army, because it was right at the start of the war, and I didn't know, I, I never really knew him, because he was killed by a V-2 while he was actually on a site in, in London. So… And that was in 1944. So, we then, my mum then struggled on with me and my younger brother until after the war had finished, and she had to work. My grandmother lived with us, she looked after my brother and I while Mum went out to work. And, she eventually met somebody else and married, so we had a stepfather.

What did your father do before the military?

He was a gardener. He was, he, he tended gardens in, you know, in these large country houses dotted around the place. So, it's an interest that I have, very much so, pick up on, and I'm very keen on gardening too. So, the genes are obviously there.

And when your mother went out to work, what did she do?

Oh well, in those days she did all sorts of things. She, she was a bus conductress; she worked in a café; she… You know, nothing very… You know, we were a, we were a very, a poor family, we didn't, you know, there wasn't a lot of money coming in. She just, she had no skills other than the sort of, workaday skills.

[03:08]
Was she keen on your education?

She was extremely keen on my education, and she, she pushed me no end. I went to the local, the local juniors, infant school, and, she was pleased when I… I remember getting a threepenny bit off one of the teachers, because I used to, I was able to read at a very early age the local newspaper, and I had to stand up in the front of the class and read this newspaper, and the teacher gave me a threepenny bit. [laughs] I was delighted.

That was quite some money.

Yes. In those days, it was. [laughs]

Yes. Did you enjoy school?

I loved school, yes. I loved the… The infant school and the junior school, I did fairly well in the junior school. Sat the Eleven Plus, and went to the local, local grammar school, which was King James I Grammar School in Bishop Auckland. Initially, it didn't quite, it didn't quite fit with me very well, and, when I was in my second year there they had to call my parents in, and, my mum, my mum in, and say that if he didn't pick up, he was going to get chucked out. The problem was, I, I couldn't get interested in a lot of the material that they were teaching us. You know, they would, in physics and chemistry it was really, not very interesting stuff, and I, I didn't really get on with it. I was more interested in playing chess in those days, and at the age of, fifteen, age of fifteen I got to the finals of the Durham County Junior Chess Championship, and lost in the finals. So I'm a very keen chess player.

That didn't show them that you had ability?

Pardon?

That didn't show them you had ability?

[hesitates] It didn't, no. No, it didn't directly. They were more interested in, you know, the academic side, and chess didn't really come into that. Because academically I wasn't, I wasn't doing particularly well because I wasn't, I wasn't essentially interested in some of the stuff that I was being taught. And, I, I picked up a couple of really, really good mentors at the school, the grammar school. The history teacher was a very good mentor for me, and he got me very interested in, in history. I took my Eleven Plus, and I got eight – or, GCE as it was then, yes in my GCE as it was then, and I got passes in, fairly good passes in eight subjects. And, science was really, started being, you know, interest me big time, and I would go along to the local library and pick up some of these advanced books on, on, you know, physics and stuff, and, the sort of stuff that they didn't teach you at the school. And, and took it on from there.
[06:21]
I had to leave the, I had to leave the grammar school, because I couldn't go on to do my O Levels[sic], you know, we needed some money coming in, so I had, I had to leave the grammar school after my, after my O Levels, and, and took up employment in a company called Bakelite, which, which were manufacturers of polymers, and, that was an interesting job, because it was…

Bakelite was a type of early plastic wasn't it?

It was. Bakelite. But they were manufacturing things like, PVC derivatives and polyethylene derivatives. So I, I got very interested in that. And in the meantime I was, I wanted to continue my education, and I, I took up, I was going to, [laughs] found myself doing, trying to do a degree with my one, they let me off for one day a week, you know, to, to go to, to the local technical college, and two evening classes. And that was quite hard. And, fortunately, I was doing so well that the, the person who was head of the science department, I was doing this at Teesside, the head of the science department wrote to the, to Bakelite, and suggested I get a, they give me a

sandwich course, because it was such a good student. And, I got that. So, what that entailed was doing six months of the year education full-time in the college, for which Bakelite paid the expenses, and then six months at work. And I did that for like, four years. And in the end…

How did you get the job at Bakelite?

[pause] I just wrote to them.

Right. And you got the, you got the…

Asked for a job. My father worked there on, on the, in the plant, he was just a, you know, working on the plant site. But, I'm not sure how much influence that had on me getting a job there.

You said your father.

Yes. I… I don't know, maybe I showed a, a level of enthusiasm that most other people didn't. And I had read this book on, on polyethylene and some derivatives of it and stuff now. I went along for the interview, and sort of blinded them with the stuff that, the fact that I had read the book, [laughs] but none of the people who were doing the interview had. [laughter] Which I thought was highly amusing.

[09:06]
So you joined Bakelite in 1956.

That's correct.

There were 270 computers in the world in 1956, and you, you stayed there for eight years.

Yes, something like that, yeah.

And gained your, and gained your first degree as a result of going on a sandwich course.

Yes, it was a… On a sandwich course. I graduated as a Member of the Royal Institute of Chemistry, with Honours. And, that was quite hard work. It was far more intense than, than I see, for example, my son [laughs] doing in university, or did at university. It was hard work. Every day there were lectures, not once a week but every day, all day. And that was, I took chemistry, physics, maths, organic chemistry, inorganic chemistry, physics and maths. And those were the subjects I studied for that degree. So, at the end of that… And during that time I was awarded the ICI Science Medal, which was a, which was a medal that was awarded annually to the best science and technology student attending the, the college at that time.
[10:31]
After that finished, I, my organic chemistry teacher, who was a huge influence on me, the next stages of my career, said, 'How would you like to do a PhD?' And, I hadn't really much idea what that was all about. So I said, 'Well, sounds interesting.' So he knew the, Professor Musgrave from Durham University, who was the professor of the chemistry department. And, he, he wrote me a letter saying, you know, 'Come along for an interview.' So we met in a pub in Durham, and, chatted a little bit. And I was offered a grant from the Imperial Smelting Corporation to do three years doctoral research in organofluorine, heterocyclic organo-, heterocyclic fluoro organic compounds, which I knew absolutely nothing about at the time, didn't even know they existed. Well in fact they didn't exist, because, fluorine is not a naturally-occurring element in hydrocarbon, so, it was all synthesising completely new, brand new chemical, brand new chemical structures, and, so on and so forth. So, I said, 'Yeah, I'll give that a go.' I got a grant, and set out on this. And, it was, I was absolutely fascinated by this whole thing, it was just such a wonderful experience. I'd just go into work in the lab, and, and a weekend would go by without, you know, without, [laughs] I hadn't noticed it was, it was Monday morning. We had local food dispensers you put some money in and got it out, got the food out. But, but I was completely taken by that.
[12:20]
And at the end of that, eighteen months, after eighteen months I had enough results to write my PhD. And it was for a year course. So he gave me some additional things to

do, which was, you know, which, they were quite successful too. So, I ended up with a thesis which was 350 pages long. And, it was during that time actually that, computing came to the fore: well not to the fore, but it crept into the subject. Because, when you synthesise these organofluorine compounds, and you, you perform reactions against them, you have no idea what the end result is, because, you know, it's brand new stuff, nobody's ever done this before. So you're trying to study these things, and see how they react with various, various reagents. Which causes changes in these things. I had no idea what these changes were.

Right.

So, I had to use, the primary weapon I used was nuclear magnetic resonance spectroscopy. Because fluorine is a, and hydrogen, are two elements that react very strongly in electromagnetic fields, and they absorb dangerous radiation. And so from that you can deduce roughly what the, what the structures of the new, resulting new compounds were. But to do that, there were some ambiguities which, which had to be resolved, because it could have been either this or that.

Mhm.

[14:00]
So I then chatted to another, one of the readers there, and he suggested that I try and work out the electron densities of all of the carbon atoms in these molecules. And that would tell me roughly where the fluorine atoms all were. But to do that, I had to, I had to use a computer. Because it involves solving universal [inaud] value problems. So, [laughs] after a struggle, I wrote a program which did that. And they had an old Elliott 80-, well, it was, it wasn't old at the time I suppose, they had an Elliott 803 computer at the university, and they had a KDF9 at the University of Newcastle which you could join to use. So, the programs ran successfully, and I was able to pretty much work out what the structures were purely from the NMR spectra that I received. And to back that up, I wasn't going to just rely on that. I had to do some… Well I had to break those new molecules down in certain ways, like with oxidation, to something simpler that was a known, existing fluoro compound, and that verified in

fact, eventually, that my, my calculations and my assumptions about what had happened were correct.

What program language did you use?

I used ALGOL 60.

Right. Did you find it a help?

[pause] I… I thought it was pretty straightforward, after I had gone through it. I was… Actually, I was helped by an X-ray crystallographer, who, I was in digs with three physicists, and one of them was an X-ray cryst-, he was doing X-way crystallography. And he, he had worked on, on a computer, and written programs for his work in X-ray crystallography, so he was a huge, he was a huge help at the time. So, essentially that was, my PhD, particularly the, the last year of it, was involved in doing… I was much more interested in computing, it was a fascinating subject, and it was a struggle, you know, chemistry, I love, and this new thing that came up, I also loved that too, and I, I was then sort of torn between… I didn't really know at the time that there was a, an industry in computing. I didn't really know much about it. [16:37]
So… The guy who ran the NMR machine, a guy called Jim Enshant, at the end of my three years he was about to move to Southampton University, to take up a post there. Because they were getting a whole new load of brand new equipment in, new spectrometers and stuff, and… So I thought, well, why not? An extra three years, I don't have to find a job now, so, [laughs] just continuing what I'm doing. So I moved, I applied to the DSIR, as it was then, Department of Science and Industrial Research, and I got a grant. I told them what I wanted to do. What I wanted to do was… There was this, just a new idea where you could take… When you do NMR, you put the compound you're studying in solution, and when it's in solution the molecules are tumbling around all over the place. So a lot of the interactions just average out. So, there are things that are missing in the spectra that you receive. The idea was to use liquid crystals, which provide an anisotropic medium, so the molecules are… Because these liquid crystal structures, they're held much more firmly in place so they can't, they can't quite move as much. So, from the spectrum,

the spectra you recover, you get much more information out of that, so that enables you to study much more complex molecules.

Right.

[18:09]
So, I did, I got enough money to spend, to do, spend two years on this. But during that time of course, the interest in computing grew ever stronger, because, again, to analyse results you still had to use computing programs to solve some of these inverse [inaud] these problems. And I programmed stuff again, it was all in ALGOL, and I used to go up to the Atlas 1 machine up at Harwell and use that for some of the, the more complex problems. But they had a local ICL machine down at the university which was a 1904, and, I got, I got well in with the computing department there, and in fact, they hadn't, they didn't… It was mainly, the work on that was mainly batch-oriented, so you put stuff in and waited for the results to come back. So… And it was pretty clunky, the system they were using. So they were trying to write a new, a new piece of software, to beef up the operating system, which made it much more amenable to screening. And I got involved in, in some of that indirectly. And so, at the end of the, of that two years, I was, you know, I… And I, I was really into computing, big time, and I was spending more and more and more time on the computing side studying, studying, like, how disks work, and all the, you know, the, the low level detail, the computing technology.
[19:51]
And at the end of it… I got a couple of papers out of it, on the chemistry side. And at the end of I thought, well I've got to, I've got to get a job now. And, I tried to get a job, I think it was Reckitt and Colman, they wanted a person to do NMR for them, you know, the… And I applied to that, and, for that. I got into the interview, and, I wasn't very keen on what I heard, the job seemed pretty, you know, narrow in terms of what, what they wanted. So… At the time there were loads of advertisements in various papers and stuff, and journals, for people to work in computers. So, I made… I thought, right, I'll have a go at this. And I made, I think it was ten applications, ten applications to various companies, [laughs] and I got ten responses saying, 'Come and have an interview.' I went… Some of them I wasn't very keen on when I got there. I mean, in fact the one with IBM, I walked out, I left halfway through the interview

because it sounded pretty, pretty, pretty awful to me. And the one that attracted me most of all was the one at Ferranti, Digital Systems Division, in Bracknell.
[21:23]
Now, Ferranti had a more, had a place up in Manchester which was the more commercial side where they were building systems for commercial use, but the Digital Systems Division worked on, on real-time systems, which was something completely new to me. I thought, well that sounds interesting. And as it turned out, it was building weapons systems platforms for the Type 21 frigates as they were at the time, and the Type 42 destroyers. And the computers were fascinating, they were absolutely amazing. They were… I had never seen anything like… They were three-address, what they call a three-address instruction code, rather than, you know, the one-address. You could do three operations at the same time in one cycle. So that meant they were very, very powerful. So, the technology itself was fascinating.

[22:29]
Now Ferranti… This was 1969.

Yes.

Ferranti had been formed in the late nineteenth century, as an electrical company.

That's correct.

And the Ferranti family…

Yeah, Basil de Ferranti, yes. Mm.

…still held some shares in it, and management of it.

Oh… Oh yes, absolutely.

Basil de Ferranti, for example.

Yeah, that's the, that's Ferranti.

And Sebastian.

Yes.

Tell me, what was Ferranti Digital Systems like then, what was the, as we would now call it, culture?

Oh, it was just like being at a university. It was just like being at a university. In fact when I started… The interview process was fascinating. The… They gave you, they gave you these tests, and they gave you four things to do. And, they were interesting things like, you had to perform the operations of a central processing unit but do it manually. [RS laughs] Or the operations of a, of a stack, but do it manually. And I remember one thing, you had this enormously long bracketed expression, and you had to remove all the brackets, and end up with a, a single number at the end of it. But what they did was, halfway through, they'd come and drag you out of the test, to give you an interview with somebody. So you'd be halfway through factoring brackets out of an expression, and then you had to, [laughs] you had to go out and talk to somebody. Then you'd have to go back and try and pick it up. I'm sure… And this was, this was done purposely, on purpose, by then.

Oh yes.

And I remember, when I did my bracketed expression, I ended up with two left brackets and one right bracket. So… [laughter] So, it didn't work too well. But that didn't matter, I mean… So, that was great. But, there were a lot, when I started there were a whole group of new graduates starting, and we all sort of, then went through a training process together, and we all lived together. There was, it was such a, it was just like being at university, but a bit more intense.

Was it good at training, Ferranti?

Oh, the training at Ferranti set me up for the rest of my career, in that, because you were working very close… You were working with an autocode, you know, where

you were, you were manipulating registers and things, directing instructions on the, the lowest level of the computer directly. And, you got a, I got… You know, you got a background in all the details of how these machines work.
[25:02]
So I was tasked with working on the operating system, the basic operating system for these things. It was a 24-bit… It was a 24-bit word machine, not, and so it was quite an unusual word size. And, you had to write the operating system in, around… It was a 16K machine, so you had to write the operating system in about 2k words. And this was dealing with, you know, lots of real-time devices, storage management, multiprogramming, all that kind of stuff. Storage management. So it was a difficult task to keep it down to that, to that, you know, number of words.

Who had chosen 24 bits for a word, and why?

I have no idea. It was just…

OK. [laughs]

No idea why they did, but that's the way… I think that's the, you know, the, how the machine evolved. I'm sure it came from a design that they had probably had for, on the commercial side of things, but I wasn't too worried about that. I was more interested in dealing with the problems that were… You know, that was something you couldn't change, it was, that was the way it was. They were very fast machines, you know. I mean obviously they had to be, because they were controlling radars, tracking targets, firing guns, firing missiles. So, it was… You know. So I worked on the operating system bit, and then there was a whole load of other people working on the weapons platforms, programming, programming for us. But, it was amazing.
[26:45]
And then… At the same time I was given the job of creating a, like a time-sharing system for them, on this machine, so people, so the various people who were doing the development could have terminals directly connected at the computer, and do their own thing on that computer, the same as several other people. And that was a very very difficult task. And I mention… I remember, it had its baptism at the time that Ted Heath chose to implement the three-day week. So they were trying to get as

much done as they possibly could in three days, you know. So, we baptised it during, during that period of time. And it, and it worked kind of OK, yeah.
[27:37]
So… So I worked for them. And they eventually decided they were going to move to Wales, Cwmbran in Wales. I think it's just, the site they had in Bracknell had outgrown its, the space it had. So they decided they would move somewhere else, and I think they got a grant from the Welsh government, they got a, they got a place down in Cwmbran. Now, I wasn't particularly keen at the time on moving to Wales, so, it was a decision time. And…

You had been there seven years?

Yes. Yes.

And what… How much were you being paid?

Oh gosh. I can't remember. Not a lot.

[28:25]
Did you have a management responsibility?

Oh yes, I did. I mean I, I managed a team, small team, which was, which was growing at the time I left, because people who worked on the, on the multitasking time-sharing systems, and…

What was the Tony Storey method of management?

[pause] Tony Storey method of management was, it was fostering teamwork. Everybody was equal. You know, there was no hierarchy. It was just encouragement, being enthusiastic. If people had problems, they could come and talk to me. I couldn't guarantee to solve the problems, but, but at least I was there to listen to what they had to say. And I listened to their input quite a lot. You know, some people could… And there was one guy in particular who was very good, he would come and say, 'Can't do that. [tapping sound] This what you need to do.' So we had a long

chat about it. 'OK, you're right, that's what we need to do.' So it was very much listening to what people had to say. And encouraging people to do that. And it works fairly well. But I have to say, I, I hate the idea of being a manager now. It's not my, it's not really my scene, I'm afraid. And…

[30:00]

So, at the time they went to Wales, I decided we were going to have to part company, and, I had no idea what I was going to do. But I saw an advertisement in the *Guardian* one day, and it was for somebody to go and work at IBM, at the Scientific Centre in Peterlee. Well I came from the north of England, so I said, 'Ah, this looks good. An opportunity get back up there, and meet some old friends.' And so I, the guy who headed it up, interviewed me, again in a pub. A common theme. And, they offered me the job, after about, you know, half an hour interview, couple of pints, and that was it, they offered the job. And the job was in, it was completely different, it was in database technology. Again, something I knew not an awful lot… I had come across it, but the, but, in the network database field, Ferranti were thinking of picking some of that stuff up and using it. So, it was in the emerging field of relational database technology.

This is 1976.

Yes. The theory had been proposed by Ted Codd, who was an Englishman, who escaped to California, and he proposed this theory, this set-oriented theory for managing data. And, the IBM Scientific Centre was a small place up in Peterlee, and, they were tasked with the idea of trying to develop a language to express the relational, various relational operations, and a system that you could run, store the data, retrieve the data; did things like optimisation. And I was particularly interested in the field of multi-user, trying to synchronise multiple users across databases at the time. So, so I worked on that. And, again, that was a, another kind of, academically-oriented thing. Seems to be a recurring theme. It was free and easy, it was more like a family than a, than a, you know, than a workplace. So, I worked on that and developed… I was working, I developed the system to, to allow queries against large quantities of data to happen efficiently. And it was also trying to extend the language to allow for certain new algebraic operations. And we, we built, we built a prototype, and it was called the Peterlee Relational Test Vehicle, PRTV. And, we tested it

actually in, we tested it in a number of places, and the big test was, was a joint project with the World Health Organisation. The World Health Organisation had gathered large quantities of statistics and information, from all over the world, about, about mortality and morbidity statistics in various, sexes and, and age groups and so on and so forth. And very often they would get queries to look at, analyse this data, you know, queries, how many of this did this, in, in, over this period of time. And, of course relational operations pretty well expressed, you know, the desire to do that, because it was a set orient. So… And they had huge quantities of this data, and the only way they could do it was by writing batch input to a program that would run, and then retrieve the data, and then, eventually get it back and use it. So we tested this system directly against the databases, against the data, to allow you to sit at a terminal, and type in a query, and get the information straight back. So that was a step…

Was this language one that a statistician could use, or did they have to have a programmer?

No, it didn't have to have a programmer. It was a… It was a bit like, a high level language interface, anybody could use. Yeah. Like, it's a bit like, do you know SQL?

Yes.

It's… Yeah, it's a, it's a, it's a bit more mathematical than that one, but it, but it's kind of easy to understand, you know, it's like, select this from that, over, you know, where this is, you know, select X from, set Y, between A and B, where A and B was some time frame [inaud]. So…

Now this is 19… Sorry. Go on.

So… And that worked. We also had a lot of interest in it from agricultural institutes from around the world, like, people in South America studying, you know, breeding of potatoes, and, some other people studying wheat. And they had gathered lots of statistics that they wanted to analyse. So, that was… So I spent time going round

giving lectures and talks to various conferences and, customers, and stuff like that, and so on. It was a good time.

[26:18]
What was the culture of IBM at the time? IBM is in its, mid-Seventies, 1972.

Yes. Well, we… It didn't show through so much to the Scientific Centre, because it was, actually, it was kind of like, separate from the mainstream, you know, from the mainstream IBM I would say, and it was well out of the way. It was up in the north of England. Although there were IBM offices in Newcastle and Manchester and places like that, who were sales-oriented for the mainstream products that IBM was making, you know, the 360, and the various software that went with that. This was kind of like a, I would hesitate to call it a skunkworks, but it was, [laughs] bordering on that. But it was, it was, it was, you know, everybody believed at the time that what we were working on was *the*, you know, *the* future for database technology. And, the guys in IBM Research in Almaden were working on System R at the time I think which was very similar, similar kind of thing.

Does it surprise you that relational technology in databases has lasted so long?

[pause] No.

Because?

It's… Because, it's, it's… I think it's something that's extensible, you know, and, and you see various people have produced, you know, various extensions to it over time, and morphs into something slightly different, with a slightly different focus, but it's still a relation-, a set-oriented, you know, relational technology under the covers. And they put all sorts of stuff on the top, and you know, enhance it in certain ways, but fundamentally it's still relational database technology.

[38:23]
The year after you joined IBM, there at Peterlee, they formed Oracle. It wasn't called Oracle then, but…

Yes.

Oracle was born. And Ingres and Oracle were, were fighting it out. But you weren't that commercially-oriented then?

No.

In 1981, IBM….

IBM… I have to say, that IBM was pretty slow on picking up on this relational technology.

Yes.

Largely… [coughs] Excuse me. Largely because, they had a database product, it was called IMS, which was, like a hierarchical database technology. And they had lots of customers using that. Now the one thing that IBM doesn't want to do, or didn't want to do, and in fact to some extent never did, was to eat its own children. So they were very slow in pushing relational technology into the market. In fact they were way behind Oracle, who got, you know, the first implementation, commercial implementations of relational database technology out to customers. And so they were then running to catch up. And they eventually came out with Db2. But, they weren't… It was a fight all the time. Every time, you know, you mentioned the word relational, you had the IMS guys jumping all over things on the computer.

Yes. Yah.

[coughs] It was a culture war.

But they also implemented it directly into System/38, and then the very successful AS/400.

Yes. Yeah, well, that was a, that was a… That was, at the time IBM was doing its, you know, its Future Systems effort. And so, you know, the underlying technology with the computers was very, you know single store, single level store. It was part of the, almost part of the instruction set of the computer, so it was completely different, different thing.

[40:34]
Then in 1981 you moved to this very important laboratory that IBM had opened in Hursley, in Hampshire.

Mm. Correct.

And you started working on the online transactional processing system…

Yes.

…which in my opinion is not talked about enough…

No, I agree.

…called CICS, Customer Information and Control System.

Correct.

And this allows secure transactions, and proper transactions, to occur. And if they don't, the whole system is rolled back to where it was before. So it's a very very important part of the guts of systems, which relieves the ordinary application programmer of a tremendous amount of, of complexity. Have I described it correctly?

That's… It doesn't… Yes, it doesn't necessarily roll the whole system back. It rolls a particular transaction that you are performing. So any changes, if it fails and doesn't complete, it rolls the effects of that specific transaction back to where it was before it began. So it can be repeated. So the whole question, the whole issue with data integrity, ensuring that the data was reliably maintained in a consistent state. So,

yeah. I mean, it was basically a, a large level of software, a large amount of software which sat on top of the operating system, which did the basic things like, running the CBU, you know, running the devices that are attached to the, the thing. But it did very little to help all these, you know, the nuances, where you wanted to manipulate a particular piece of data, or collection of data, it just, didn't allow you to do that. So we, we built this thing. It was initially built, actually, as a, as an application for, Florida Power and Light was it? It was for some power company in the US.

A utility. Yah.

A utility company in the US. And then, and then, eventually, after, it was transferred into Hursley in exchange for, I think we sent VTAM the other way. And the idea was that this thing… It was a bit of a, a, a bit of a hassle for them, and they moved it to Hursley I think with the intention that they would ultimately die. [RS laughs] Little did they know. But it, it has become, you know, the de facto standard for online transaction processing in the world. And it, you know, you can talk about all these, the number of transactions, the number of interactions that go on on the Web in a, in a given period of time, and you can talk about all those things, but the number of transactions that it reliably executes in the same time is in orders of magnitude greater. And it's used by enterprises all over the world to run their businesses. And so it has to be reliable, it has to always be there. It has not got to fail, and if does fail, it's got to be able to recover quickly, and with some level of integrity bestowed on it.

If you go to the cash machine and you try and get out your money, and it says, 'Yeah, you can have the money, you've got enough,' so you, you want your £100, the £100 is taken out of your account, in the central accounting process, but the cash machine screws up the money and can't, can't get it out because it's crumpled or something, you want to be able to stop that transaction and roll the whole thing back. And basically, it's that type of integrity in transaction processing that CICS gives you.

Yeah. And other things, like, if you have several people, if you have three people all manipulating a, a given dataset at the same time, you want to be sure that they're not interfering with each other, you know, otherwise the, you know, the data gets completely messed up.

[44:53]
And again, Tony, you're working extremely close to the hardware, to get these high levels of transactions through as fast as possible. You've got flying heads spinning across: well they're not spinning, they're flying, across disks, picking up information.

Yup.

Processors processing it as fast as possible. And you're pumping it out through lines. So again, you've got to get right down, very close to the hardware, to make sure you've got no real clunkiness of that software.

Well, we don't get down, we… You actually don't get down that close to the… All the, all the hardware is manipulated by the basic operating system.

Oh sure, yeah, yeah. Yeah.

But we have to ensure that what we get, you know, the information that we retrieve from these various devices, has a level of consistency and integrity. We often use the word ACID: atomicity, so, you know, everything behaves as though it were a single operation, consistency, integrity, and durability. You know, so if you make a change, it lasts.

Yes. Yup.

Isolation is, you're isolated and you're in your own little word, and reliably isolated from all the other people who are using.

[46:07]
Now this piece of software, CICS, made millions and millions and millions of dollars of profit for IBM.

I would say billions of dollars in revenue for IBM.

And it not only ran on IBM's proprietary systems, the 360, but you might well have been involved in its transfer to open systems, and to users.

Yes, I was in fact. I, I, with, you know, a lot of initial resistance, decided I wanted to put, it was a good, why didn't we put this product on the RS/6000, which is a RISC computer that IBM came out with, and it ran Unix, or AIX as it was previously, it was the Unix operation system basically. And so there was a… I had decided I would like to do that, and, again, that was a bit more of a skunkworks. And I got permission by my manager at the time to give it a go. And so I, we, we spent some effort and time, got it running on the RS/6000 under AIX. Now, AIX, being a Unix system, allowed us to put up that program. I mean it was a complete reimplementation of the, of CICS, exploiting some of the specific hardware features, like the segmentation, memory segmentation [inaud] on the RISC [inaud] 6000. And, we were able to take that and transfer it to other platforms. So it ran on HP, it ran on SUN machines, it ran on DEC machines, and… So there was a lot of umping and ahring [laughs] about, about that. And that was called… It wasn't called CICS; they eventually called it TXSeries, but it was basically a reimplementation of CICS on a Unix system.

[48:01]
Right. Now, there's one thing that I'm going to, we've got to press on a bit. I want to ask you some contemporary questions and some historical questions, before we go on to your, to your later work, if you don't mind.

Mhm.

What is your reaction to the news over the last few days that several thousands of exchanges, mail exchanges, from Microsoft have been hacked?

[pause] Well I'm, I'm pretty well appalled about that. You know, I, I've never seen Microsoft as a paragon of, you know, of reliability, and that's shown by the number of updates you used to get. But I'm, I'm horrified. But, you know, these people are very clever. And, if they find a little, a way into a system, they'll use it. [pause] I mean, you've just got to look at what's going on. I mean there's always, just recently there's been, I think it was on the news today that China had been responsible for hacking a

lot of businesses in the US. I think, you know, you have to… I don't think we pay enough attention to this whole business of cybersecurity. It needs a lot more effort on it than is currently being put on it. But, at the same time, you know, the enterprises who are using this stuff themselves have to… Because a lot of the, a lot of the, the ways in are through the actual software that some of these companies are actually writing, more than anything else. It's bad. It's very bad. I mean, it's giving IT a really bad name.

[49:53]
The… Watching software evolve and change over the years… Don't take this wrong, because I know you're personally devoted to software and software engineering and very enthusiastic about it. It does seem to me often an area of fads and fashion. Does it strike you like that?

Yes, it does. But these things come and go.

Right.

The important thing is just keep going, on and on, you know. And you have to try fads and fashion. You know, people need to sell software, you've got to get people to buy it, you've got to get people to use it.

Yah.

And some of the fads, you know, in the past, people have tried fancy things and lots of different things, and some have… And a lot of them, most of them probably, have fallen by the wayside. I think that's the way the industry evolves. You try new things. You try them, hey, they work. And so you work more on them and get them, and they become better and better. Others just, whoever thought of this, you know, it's rubbish, and they just fall by the way. And, I, I… You know, that's, that's life I think in the IT industry, it happens all the time. And there's nothing wrong with it. I don't think there's anything wrong with it, providing it doesn't cause major problems. Because that's the way things evolve.

[51:20]
Do you call yourself a software engineer?

[pause] I call myself… I've got to call myself something. I'm more a computer scientist I think.

Right.

Well no, I'm never one for titles, but, software engineer is a job title.

It always strikes me that it's a bit like political science.

Yup. [laughs] Yes. [laughs] It'll do. [laughs]

[51:51]
It'll just about do. One of the fads was, message-based systems.

Ah yes,

And you went on to propose and develop the IBM MQ series of message-based. But can you explain that please, and when that was?

Well it's, very simply it's, it's a, it's a way of delivering messages. It's two things. It's a way of delivering messages reliably to another, to somewhere else, between two, messages between two programs. But the important thing is that the program, the recipient doesn't have to be around at the time. So it can be like, delivered later on but still reliably. So that, you know, the message is managed all the way through the cycle of being sent all the way through all, to somebody at the other end who receives it. And it's also quite importantly, a new, you can call it, almost call it a new programming model for building applications, where you can take an application that exists for example, and build a new one, and connect… You don't have to modify this, but you can connect this other thing to it, through this message-driven processing technology. And it's, the whole process is managed. It's reliable, guaranteed. And, you know, it sounds simple, but, there isn't much software around that does that, or

there wasn't much software around that did that. Simple ideas are probably the best ones.

It didn't get as much popular support, this approach, as, as CICS did.

Oh. It's… Certainly not, but there's more people using, more people building on this one. So… So, it's, it found a foundation for it, for other things, like message brokers, and, all sorts of, you know, other derivative kinds of software built on the base of message queuing. So I think, it's probably just as widely used now as CICS.

OK. You…

And its derivatives, you know, people build, you know, smaller versions of it, real-time versions of it, you know, and those variations on it, plus the thing itself, probably add up to a significant amount of use.

[54:28]
You did a lot of work as well personally at IBM to turn Java from just something into something very big, did you not?

Yes. Java came out as a language, just as a language. You know, it was a programming language. And, I helped, with some other people I have to add, and not all from IBM, but from various other IT manufacturers, but we worked on developing what we called an Enterprise, an Enterprise Java infrastructure. So we built a, we defined a component model for Java, Java components, and, you know, various specifications that cover what, the, the types of transaction sys-, the transaction processes, the online transactions, the messaging technology, the data technology and so on and so forth, which allowed, which were necessary to build enterprise applications with, at that level. So basically extended the language to, to build this higher level infrastructure that allowed construction of business applications.

And by this act, IBM is, is part of IBM moving into open systems, is it not? From its purely proprietary approach, you are not going…

It was then, yes it was, in a sense, working, working… It was working on international standards that allow, you know, systems to work with each other more effectively than they had done in the past. So it was, they were international standards, not proprietary interfaces or proprietary protocols; they were all out there in the public domain.

[56:21]
And this brings me to a characteristic of yours, which I'm going to suggest to you, you can reject it, you are a bit of an iconoclast, are you not?

[pause] Well, I… [laughs]

Well…

Yeah, I suppose I am.

CICS… CICS, you can put it on Unix. Oh really? I bet there was resistance to that inside IBM.

Oh a huge amount of resistance. [laughs] And to put it on other platforms? No, you must be joking, sir. [laughs]

Java? Oh we're going to have Java. Oh are we? What's so wrong about C? We've been using it for so long. No, it's going to be Java.

Yeah, or, or, C or PL/1 or PLX, or…

So there's this strand…

So, portability. You know, the big thing was portability, interruptibility, all these things that have become important.

Right.

So, I suppose in a sense, that's probably a fair classification, iconoclast.

[57:21]
What's the biggest mistakes you've made in your career, and what have they taught you?

[long pause] My biggest mistake, I have to say, is being, too dogmatic. [tapping sound] This is what should be done.

Where does that come from?

I don't know. Because it… [pause] I don't know. It's the need to, I suppose it comes from a need to succeed, and not trusting other people, which is a bad thing. But yeah, I think, I was probably too, probably too forceful in my approach to making things happen.

You think you have to be?

I think you have to be, yes. But, you can be much more, probably much more diplomatic about it than I often was.

[58:48]
You didn't leave IBM, did you? You stayed with IBM through your…

I stayed… I stayed… I, I stayed right to the end. In fact I stayed beyond the end, because I retired on, in 2004, my 63rd birthday. And then I… 2000, 2004. Yes, 2004. And then, I stayed on for another, five years. 2009 when I finally finished.

Did it ever occur to you that Tony Storey could have broken out, and with these kills produce a commercial product which could have made you a multi-multi-multimillionaire?

[pause] No.

Why not?

I never, I don't… [pause] Well the things I was working on at the time, there was probably, a lot of heavy-duty competition. So getting something started from the ground up, would have been an extremely difficult operation. And also, I was actually enjoying very much… They were paying me enough. You know, I'm not, I'm not a man who wants to be a millionaire, although I, [laughs] I do have a lot of money now. But, you know, money doesn't worry me, as long as I've got enough.

It's more the technical challenge, is it?

Yes, absolutely.

[01:00:22]
Where would you advise young people to go into this industry now, Tony?

Where. Well my son's gone into the industry, and he's working for Roke Manor, who again is a defence-based, defence business company. I think probably one, one of the… I think if I was going to do it again, I think I would go into the whole cybersecurity space. There's a huge challenge there, and a huge, and huge, you know, huge opportunities to, to make your way in that. I think that was… And he's trying… Actually, he is trying to do that at Roke, get into the, part of their company that deals with cybersecurity. Which is obviously very important for the military.

[01:01:20]
Does bit chain technology excite you?

Sorry, bitcoin?

Yeah, bit chain, bit chain.

Oh…

Blockchain, blockchain.

Blockchain.

Sorry, blockchain.

Yeah, it, it's… It doesn't… It excites me, but, I'm… It relies… It relies on a lot of processing power to be effective. I know IBM has, is doing some work on it, I know one of the guys who is actually working on that. [pause] I'm not sure whether it'll have huge success, but it's obviously important in the bitcoin, you know, the bitcoin area. But it consumes vast amounts of, vast amounts of processing power to use it. The area that interests me most probably is quantum computing. I think I would probably be interested in trying to get into that field if I was starting all over again.

Isn't that something that's always going to be coming?

It could well be. IBM has a prototype, research lab up, that I use. I've been fiddling around with that just a little bit [inaud].

You have been?

[laughs] Yeah. But it's, but it's, it could be one of those areas that's always coming, but, you know, you, you've got the opportunity to make it work, you know? It's one of those fields where you have the opportunity to make it work, and make it big.

And that seems to me what turns you on, making it work.

Yeah. Exactly.

Well thank you very much Tony Storey for making this interview as part of the Archives really work today. Thank you very much indeed.

OK. OK, it's a pleasure.

[End of Interview]