



Martyn Catlow

Interviewed by

Richard Sharpe

15th August 2022

Via Teams

Copyright

Archives of IT

(Registered Charity 1164198)

Welcome to the Archives of Information Technology where we capture the past and inspire the future. It's 15th August 2022 and it is officially a drought in much of south England, which is where we are at the moment through the wonderful use of the internet. I'm Richard Sharpe and I've been covering, analysing, teaching about and pondering the IT industry since the 1970s. It's quite appropriate that we're going to have three contributions at least from the Meteorological Office, a centre of enormously important computing expertise in this country and a long and valued history of it. And making his contribution to the Archives today is Martyn Catlow, whose avowed professional passions are weather and computing, and so weather and computing, he'll be able to talk about. Martin, you were born in Blackburn, an industrial cotton town, what were your parents doing?

Well, my parents were not particularly well educated, they were working class people, they had ordinary jobs. My mother was a mill worker and she worked the looms in the cotton mills and my father was a delivery driver. So, yeah, humble beginnings.

And no university background or anything like that, just normal education?

I went to a secondary school after primary school, but primary school was probably the place that influenced my interest in weather the most and that led to, you know, other things later on, but I went to a secondary school called Billinge, which was an ex-grammar school, and had a really good education there because a lot of the teachers were still from that old grammar school, which was really good news. But I decided at the end of that period to get a job rather than go to university.

Okay. Let's get you born to begin with. You were born on 16th October 1960. The year before, your now employers of your three employers have got their first big computer, it was a Ferranti Mercury, and they called it Meteor, and it was one of 5,500 computers in the whole world.

Yes. That's a remarkable thing.

So, did you enjoy school?

I certainly did, yeah. And, you know, I came away with a couple of A levels in geography and physics, which was good for meteorology, and that's where I ended up, at the Met Office.

Why were you allowed to mix geography and physics? That seems to be an arts and a science mixture.

Well, as I said, they were quite a forward-looking school and they didn't silo people into particular directions, they were quite happy to allow people to make the choices that best fit their approach and needs, really.

Well, that's good. Did your parents support you in education?

A hundred per cent, yeah, and they were prepared to let me go to university. I was going to go to Lancaster University to study environmental science, but the job at the Met Office came up and that seemed a better option.

And did you have brothers and sisters?

No, I'm a single child. Yeah.

The sole recipient of their devotion.

[laughs] Yes.

Do you think that makes a difference to you? Or perhaps you're unable to tell because it's your natural state?

I don't know. To be honest with you, I don't think it makes any difference really. I think you are, as an individual, you grow into who you're going to be in your late teens and I think if you've come from a large family you may be influenced by your brothers and sisters, but I was quite directional in what I wanted to do from an early age.

You went to a primary school in Blackburn. In 1965, while you were in the primary school, the Met Office bought an English Electric KDF9.

That's it, yeah.

And they called it the Comet, and it cost them half a million pounds.

I think so, yes. But I was completely unaware of that, I was completely unaware of all these things going on, but you know, the centre of excellence for computing was just a few miles away from Blackburn, down in Manchester where they developed a lot of the technology that underpinned these machines that were beginning to be introduced...

Indeed.

You know, the LEO. Yeah?

[00:05:10]

Is Kidsgrove in Manchester?

In the south, isn't it, Kidsgrove, I think?

Okay. Because KDF9, apparently in English Electric, means Kidsgrove Factory number 9.

Yes.

KDF9.

That was, I think that was... that came out of the LEO project, didn't it?

Probably, yeah. And in 1965 you're also still in primary school, they did their first operational projection forecast on a computer, in November 1965. So you went on to

Billing [Billinge] High School, which was a progressive. Physics and... geography, was it? Yeah?

That's right, yes.

And 1970 they actually got – 1971 in fact – the Met Office got an IBM 360 195, a big system indeed. So you decided to have a look at the Met Office, did you? Why? Because it, because of this interest in weather?

Yeah, I mean the main driver for me was the weather, but little did I know that, you know, to do all the weather forecasting and processing at that time, I thought it was a manual process, but it turns out that they were introducing computers around the time where I was leaving school, they'd got this 360 195 IBM machine, which was one of the most powerful computers in the world at the time. And in fact they ran the first computer models for atmospheric simulation on that machine, very small scale of course, but they were quite sophisticated for the time and leading the game in terms of computerised weather forecasting. So joining the Met Office at that point in time, in 1979, they were really ramping up how they were using that machine, and not only were they using it for weather forecasting, but they were using it for climate data digitisation, which, you know, a huge history of climate data has been generated over the years since 1850s and that was all being digitised onto magnetic tape. And that was one of my first jobs, actually, to look after the archive, the digitised archive, and manage the tape archive that it was being put on to.

So were they actually putting onto tape the original records of that original – it wasn't called the Meteorological Office – the Met Office under Admiral Robert FitzRoy, were they really doing that?

There were two ways of tackling that digitisation. One went back to the 1920s where data from logs and ships' logs, marine data, was punched onto punched cards, Hollerith cards, and that continued right up to the 1960s, I think. But once they got the digital computers, the electronic digital computers, there was a – because of course the punched card system was a mechanical system and a very archaic way of data processing - but once they got the electronic digital computers of the 1960s and

seventies, the approach to data processing completely changed and it was done at scale. So they started to not only punch data from records directly on specialised computer systems that allowed teams of people to basically key in information from forms by visually translating it and using a keyboard, into digitised form, but they'd also got this archive back to the 1920s of punched cards which they also digitised into electronic form by using card readers and so on. So, you know, there were two channels of data flow coming into the archives at that point.

[00:09:56]

And they were using this to look for patterns, were they?

So the data, yeah, obviously, you know, going back, even back to the 1850s there was analogue computing going on at the Met Office and Professor Galton and his colleagues were developing mechanical machines to process data and do analysis of daily patterns and, you know, behaviour of climate over longer periods of time. And that was actually, you know, that early work has fed into the sort of modelling that we do today, in a very loose way, but obviously everything's digitised now so it's much easier to write more sophisticated models than it is to write programs, if you like, that control mechanical machines. So...

Now let me just put a, put my big toe in the water of meteorology and try and explain it to you, you tell me where I'm wrong.

Yes.

Presumably you get better results if, one, your model is correct. So let's just put that to a side for a moment, the model. But also, the data that you collect, one, it's the right type of data – windspeed, pressure, temperature, sunlight, those types of things, time as well - the right type and it's accurate. Two, it's in a little cell, it's in a small area. If you have the average sunshine or average temperature for the whole of the UK, that's not going to tell you what's going to happen in Brighton, necessarily. And so the more granular you can have it, the better it is. Is that basically right?

In terms of forecasting, the geographical area that's covered has obviously increased from the early days where it was very oriented towards the UK and we now do global modelling. And for each- this is very difficult to describe how a model works really without visual material to help, but I'll try and describe it. If you imagine a grid that covers the entire surface of the earth, at each intersection of the verticals and horizontal lines of that grid – and the grid spacing can be kilometres or, you know, tens of kilometres, depending on the resolution – at each intersection point you can forecast the parameters that you've described previously: windspeed, temperature, humidity and a whole range of other parameters get forecasted these days in the very sophisticated models that we run. And that's done on a global scale and in more detail in more condensed form and much higher resolution over certain areas of the world, like the UK, for example, and Europe. And that is presented in digitised form through those channels to the media and for specialist consumers like aviation and so on. And they all have different requirements. So aviation, for example, would be more interested in what's going on at the higher levels of the atmosphere as opposed to what's going on at the surface. And so we do vertical forecasting as well as the points on the surface grid. So if you imagine different layers in the atmosphere, different grids on top of the surface grid going up for each point there, you've got a vertical representation of that data as well. And that basically means there's a whole load of data and the more you increase the resolution of those models, the more data you've got to process. And, you know, the aim is to process that in a minimal time so that people get a good forecast out at the end.

[00:14:28]

So you went to the Met Office College in Reading, because it was then, it was based in Bracknell, which is close to Reading. This was your first time away from home, was it Martyn?

Yeah, it was, yeah. I left...

And you were down in the soft south as well, not up in Blackburn.

That was quite a change, actually, yeah. Because Bracknell was a new town and there was very little there, actually. The Met Office was probably one of the biggest employers there. Of course they had ICL and they had an industrial area and they had the old ICL computing, the home of ICL computing there as well.

Yeah.

So it was a bit of a hotbed, bit of a mini Thames Valley [laughs]. Not mini Thames Valley, sorry, you know, the Silicon Valley, Inner Thames Valley, if you like. And, yeah, you could see there was a lot of money being invested into that area and particularly in IT, even in the early days of the 1980s there was a clear indication that, you know, Reading and Bracknell and other towns along the M4 corridor that were going to become specialist IT centres, if you like.

How long was your course at the college?

So, my course at the college was, there were several over the early years and the first one I did was an introduction for meteorology, which I'm proud to say, I came top in that class. But yeah, that was only a month, so it was a very specialist course in basics of meteorology as the Met Office, you know, did their work and it was introducing you into forecasting and observations and all the sort of good stuff that generates data, really.

So now you have become a civil servant and your first title is Assistant Scientific Officer.

That's correct, yes.

This is when you are doing the punched card processing and getting stuff actually up onto the computers. How did you verify the information? What type of verification processes were involved?

That's a very interesting point actually, because data on its own that's been manually recorded can have error, no matter what the data is, and if you're reading instruments

manually – these were the days where they had Stevenson screens with thermometers that got read at, you know, periods during the day, 9 o'clock, 12 o'clock, you know, through the day every three hours and so on. And those readings were subject to error. So there were two real ways of ensuring the consistency of that data. One was ensuring that the difference between stations in a particular locality was not wildly out, so that you'd compare a station maybe twenty miles away from another station to see whether there was a massive variation in temperatures. And if there was, then you would ring the observer up and say, is it possible you made an error of five degrees, or ten degrees if they were reading a thermometer, for example, a scale misread or something like that. And yeah, so there was, it was a very manual process until the mid-eighties when they introduced a computerised version of that comparison, which was done by a computer program called Areal, which was basically an aerial comparison and a geographical comparison between digitised data in a particular area. And that was the start of automated data processing really, or intelligent data processing.

You did this, you were in this role – sorry – Assistant Scientific Officer, this job title, for about two years or so. Then you moved on to be a Support Programmer, Climate Data Processing was your job. What was that involved?

Yeah, so at the time the Met Office had some experienced programmers that were people that had basically come out of forecasting or, you know, basically forecasting jobs and had a career change because they wanted to get involved in computing. And they'd done that during the 1970s, they were a cohort of people that had been gathered together and trained specifically in the use of programming languages, like Fortran and Assembler, for developing programs to do this processing with all the data that was now being made available from the archives and from the forecast models that they were able to now run. And those people were already producing programs that were producing reports for commercial purposes and for aviation and data analysis for climate data and so on. So I got involved with those folks, because they were sitting just down the corridor from where the data processing that I was doing, the quality control with data, which I was initially involved with, and I obviously was influenced by what they were doing. They saw that I was influenced by what they were doing and recruited me into the programming side of things,

because I had a good knowledge of the data, I understood what data was about, and you know, there was an opportunity there to understand how you could do much more with it by using computerised techniques. And yes, met some very, very clever people that were pathfinding at that time and they were a big influence to my future career, actually.

[00:20:58]

Is this where you started your Fortran programming?

It is exactly that, yes. And yeah, some very good tutors who were very bright and understood meteorology and also had an aptitude for programming, and they conveyed that over to me and triggered things in my mind that allowed me to understand how to program. And yeah, I had an aptitude for that as well, which was great. And that's the theme for the rest of my career, really.

It is quite an elegant, the programming language, Fortran, isn't it?

It's simple compared with some of the object languages that we have today, which can get very complex to understand. But yes, I loved Fortran because it was very scientific and you could translate exactly what you wanted to do back to very simple coding techniques and they were simple in those days. Languages, all languages now have become much more sophisticated and more capable. But yeah, I really enjoyed—and specially programming with low-level language, which happened a bit later on, but yeah, that really makes you understand how computing relates to the underpinning hardware.

There was a ferocious debate in the Fortran world about the use, or the non-use, of the Go To, ban the Go To. Did you become a participant in that debate?

Oh, I've always used Do Loops and never branched out of a Do Loop.

You, excuse me - I'm glad I'm sitting down – you never branched out of a Do Loop?

I never used a Go To to branch out of a Do Loop after an If, no.

Oh, wonderful.

Did you? [laughs]

Well, yes, until I was shouted at, yes. [laughs] So, Edsger Dijkstra would love you because he was ferocious about trying to squash Go To.

Yeah. I think the trouble with Go To was, it used to create a very sporadic program flow, and not only was that bad for the machine architecture because of the way information was cached in the storage and accessed, it was also very bad programming when you wanted to read what somebody else had written, because you could never really understand, you know, when something was going to jump from A to B.

Yeah, yeah. Why did she write that, I used to sit there and think when I was trying to work out somebody else's code and maintain it. And during this period, while you were Support Programmer, did you, you went back into education, Reading College of Technology and Design, and did your HNC in computer studies.

That's right, the Met Office were a very, very, very good employer in terms of supporting people and encouraging them to further their education. Now, I'll be open and honest, I'm not the brightest tool in the box, but you know, I was really interested in learning more about computing, the theory of computing and the technologies that were becoming available, like databases and so on, in the early eighties and that was a really good opportunity that the Met Office gave people that, you know, were interested in furthering their education. They paid for training and that was a two-year course I did at Reading Tech, and thankfully I came out with a reasonable result of a computing studies HNC.

[00:24:49]

It is an exciting process, learning about programming, isn't it? There is really intellectual excitement involved in it. I know it sounds nerdy, but I felt that as well.

Yeah, I think, you know, if you can make a machine do something and see that you've made that happen, there is a real feeling of, it's not power, it's the feeling of achievement, you know, that you've mastered something. It's a bit, I guess it's a bit like playing an instrument, or a violin or something like that, you can make it and hone it to make the most beautiful sound and you can, if you're really good at programming you can make a fantastic program that uses the machine to its best efficiency and, you know, produces the very best results. And that's what you aim to do as a programmer, I think, isn't it?

I came from an arts background where everything could be reinterpreted. You could have a reinterpretation of Jane Austen, or a reinterpretation of the causes of the First World War, but when I met my first computer, I was absolutely gobsmacked by the idea, no, no, it'll just do what you tell it to do, it is a finite machine. I thought that was scintillatingly brilliant. Another thing I thought was brilliant was being shown the Shell sort, and by the end of the afternoon I'd written a Shell sort program in Fortran. I was very pleased with myself. Did you use a package for sorting, or did you write your own sort programs?

Oh, I think we all wrote a sort program at some point, particularly when we were being trained to do things. But, as I say, by the time I'd started to do things at the Met Office from a Fortran programming point of view, there was a lot of stuff that had been developed by these guys that had been working on it during the 1970s and they had produced sets of routines and libraries that people could reuse, rather than program code themselves from scratch every time, there were libraries of regularly used routines. Like, for example, to calculate relative humidity from temperature and the wet bulb. And, you know, that allowed a standardisation to start to come into what we were doing and really important stuff, actually, because at the end of the day, the quality of modelling and the quality of data or control all depends on having standards. There's no point having one routine that calculates relative humidity in a

different way to another routine and then having similar sets of data that are being, later on, being compared if you're relying on that derivative, the relative humidity, and it's been calculated incorrectly by one of those routines. So, you know, really important that, you know, it was the basis of what we have today which is, you know, common libraries of routines for open source programming and so on.

When you have graduated with your HNC, was that why you then got a promotion to Consultancy and Training Explorer?

I don't know, I don't know whether it was directly that. But clearly, you know, that was a proving ground to say, you know, what is the capability of this guy, can he do more. And I guess, being challenged in life is two things, but one is some people have a breeze because they're just intellectually, you know, completely bright, and other people have to put a lot of work into achieving outcomes, and I think I was the latter. I did things and achieved things through, you know, learning and doing rather than taking an intellectual approach in many cases, and I think the outcome of that is that you can still achieve in life, even if you're not particularly gifted in terms of academic. So you can break through and you become an engineer through practice, if you know what I mean.

Yeah.

And that can be just as valuable because you learn an awful lot as you go along that way. And I think that allowed me to move into this next area of work, which was techniques and training, they called it, where I was using some of the techniques that I both developed and learnt from my colleagues and passed them onto a new generation of people that were coming into the Office after me during the early eighties.

Praxis – P-R-A-X-I-S – being the combination of theory and practice, not separated but blended together.

I think that's it, yeah. And I think there's a beauty to that because it converges the, you know, the practicality of the real world with the academia, if you like.

[00:30:10]

Right. You were involved in quite complex graphic programming then, is that right?

I wouldn't say complex, but various, for a short period during my time I was involved with some of the early graphics stuff that the Met Office did in terms of data representation. And one of the massive challenges, we were just talking a few minutes ago about Do Loops and Go Tos in Fortran, and I was given the task of sorting out a program called CMap which was written by a very, very bright guy, Peter Cockrell. And he developed this program which basically did a visualisation of data. And it had been around for a few years, in fact it must have been written in the 1970s and late seventies and in the early eighties they wanted to refresh that program so it could use colour graphics as opposed to just drawing lines, which was, yeah, line drawing was the only capability available with the early plotting machines. But as we started to move into the era where we had VDUs that were capable of doing on-screen graphics then new tools became available. And that was one of the biggest challenges I had at a very early age, was to convert this program away from line drawing into raster graphics so that it could be displayed on a VDU. I didn't complete the job, but I started it off, and yeah, that was a fantastic exploration of someone else's very very deep and clever code.

At this time as well you were training other people. Did you take to training easily?

I did at the time. I think I was probably more confident than I am today, believe it or not, in my early, in my mid-twenties. And yeah, it was really good fun actually, training people, because you just... it was challenging, obviously, standing in front of a group of people and feeling confident about what you're presenting to them, it is definitely a challenge, but I think I got there and I think I was master of the stuff that I was conveying at that point, which, you know, you really have to be in order to convey confidence to the people that are learning. So yeah, yeah. It was good fun.

You were doing lecturing in evening classes at Reading as well, apparently?

Yeah. Well, one of my colleagues that worked in the climate data processing, he was the senior, head of department, and he'd been a Fortran programmer for many years, but he also did a lot of lecturing at Reading Tech, but he wanted to pack that in, so he suggested to me that I might like to take over from him. And in fact I did for, I think it was a couple of years. It was quite hard work though, obviously having worked during the day and then going to do a couple of hours in the evening to train people on equally challenging coding techniques and so on, it was interesting and you met a lot of different people from different businesses and different organisations who were, you know, trying to learn the same stuff that you had done. But yeah, it was very hard work actually, yeah, teaching.

But rewarding?

Yes, definitely, definitely.

I never knew why with some people it clicked and with others it didn't, when I was teaching. I was never quite sure why, but I was very happy when it did click for them.

Yes, I remember one occasion where I was in a bit of a loop with someone in one of the classes that I was giving and I just couldn't get them to understand it. So I think at that point you just have to say, let's talk about it later, otherwise you just block things, don't you?

[00:34:28]

Now, we're in the mid-eighties now. In 1985 you joined what is the IBM mainframe system team. What was the IBM mainframe, what was it used for?

You mentioned earlier the 360 195, which... and that was the mainframe, that was the IBM mainframe.

Okay.

That had, at the time I joined it was being replaced by newer versions of the same technology and that team that I moved into was a very specialist team and that was the most engineering focussed job I think I've ever done. It was low-level stuff that was making the best out of that investment in the mainframe technology. And of course at that time in the mid-eighties, the mainframe was probably the only machine that was doing significant data processing in the office, so everyone in the Met Office relied on that machine, its availability, its performance and the tools that were available to help them do their jobs in terms of using data processing. So it was a small team, a team of probably about seven to ten people, and they were very focussed on engineering around making sure that that mainframe system and the peripherals that were used associated with that, the graphical presentation machines, the printers, the tape drives, the disk drives and all that sort of stuff, low-level IT, all hung together so that the Met Office had a twenty-four by seven production platform for its weather forecasting.

What was the operating system?

The operating system at that time was a proprietary operating system which was called System 390, and it was called System 390 because 390 was the points in the compass, represent... and the platform was intended to be a multi-purpose platform so it could work in any direction, so they called it System 390. Yeah, that was the origin of that, I think. It was a multi-purpose platform and it did everything from forecasting through data management and data archiving, so we had the first of our, well, at the time it seemed like large-scale data archives, but they were only gigabits in size. Today we have petabits and exabits, indeed, in size. But yeah, the machine was the focus of all of the data processing in the Office in the mid-eighties, really.

You really got to know how that machine worked, because you were dealing with an Assembler language.

Yes, Assembler's like Fortran but it's better.

[laughs]

Because it allows you to go right down into the mechanics of the machine and operate aspects of the machine that other programming languages couldn't do. It's designed to interact with the instruction set of the machine directly, so you're handling machine operation codes, bits and bytes of storage, and changing, you know, individual bits of storage and privileged areas of storage as well, so not just areas of storage that are like a Fortran program could access, but... And doing that allowed you to access things like computer peripheral channels, communications networks and actually, because Fortran was a very – how can I put it – very high-level environment, it didn't allow you to interact with specific devices particularly. It had the concept of a card reader and printer, as you'll know, but it didn't understand screens, screen technology, and it didn't understand communications channels. So being able to program in lower level languages like Assembler enabled you to write routines that could be called from Fortran, yeah?

Yeah.

But will also interact with some of these devices and channels that were very specific to the machine architecture. And enabling that allowed data to be passed between computers. At that time we had other computers which were very specific to data exchange: Ferranti Argus and various other specialist processors. The data that was being produced on the mainframe was sent to those machines and it was sent using low-level language like the ones that I was involved with writing. And of course, it's no good holding data on the computer, you have to get it out to the people that need it, and as I mentioned earlier, we had various customers, commercial customers and in particular, aviation customers and commercial airlines used to take the data, but take the data from the models that were running on the mainframe out to those customers, you had to be able to transfer it through these channels and Fortran alone wouldn't do that. So you had to be able to write low-level programs to interact with those communication devices and push the data out to these other systems that would then spread it and broadcast it, if you like, out to the world.

[00:40:44]

Did you do any transaction processing?

And so transaction processing is an interesting one because the forecasting people did do transaction processing in a very – how can I put it – elementary way because they were interacting with some of the computer models on some of the very basic VDU screens that were around in the early eighties. So they could, what they call, intervene. If a forecast was going off because it had, if it was going off-piste, if you like, because it had some wrong data or there was a bad data point being input into the forecast and it's causing its forecast to have abnormally high temperatures, for example, then they could intervene on the screen and change those parameters interactively and enable the forecast to continue without going completely wildly out. So that was a human intervention, if you like, and that used interactive access. But largely the processing was batch. It was all about large volumes of data, large volumes of model outputs and moving that from one place to another, from where it was being generated to customers who wanted to consume it.

You created an automatic job scheduling system for your mainframe batch workloads, yes?

That's right, yeah, yeah. And that was basically, because different workloads used the same resource, so you'd have people wanting to use tape drives, you'd have people wanting to produce graphical data, you'd have a whole host of different workloads running on the mainframe because it was a very 360 oriented machine, it could do everything. Didn't necessarily do everything very well, but it did everything, and because that variation in all of those workloads, there were tools and toys needed to enable that stuff to happen. So yeah, I was involved in writing a lot of the stuff around how you managed the resource of the mainframe in an elegant way such that everything managed to get processed, rather than, you know, the emphasis being on one thing over another. And it was a shared environment so everyone had a fair share, if you like, of the computing. And to do that there was various approaches to scheduling different types of workload, if you can imagine that.

You installed remote terminal access for Edinburgh and Belfast Met Offices, yes?

Did indeed, yeah. That was a great experience. Believe it or not, in the mid-eighties the regional offices of the Met Office still had no access to the central computing, so there was an initiative to try and help that spread, that push of access out to the regions, and I was lucky enough to spend a week in Edinburgh and a week in Belfast and help them set up the remote communications and access to the mainframe. And yeah, that was a great day when they turned on the terminal and could see a program running at Bracknell and access it and interact with it from, you know, Edinburgh and Belfast.

And what lines did you use?

They were leased lines. Yeah, so they would have been provided through BT, I think, at the time, I don't recall the exact details, but they would have been part of the BT infrastructure and very low speed. You know, not like the infrastructure, fibre infrastructure that we have today. I think we're talking here about small numbers of megabytes, ten megabyte per second links. So, you know, screens, green screens with alphanumeric data, no graphics. But basic access to central computing in Bracknell from the regions, yeah.

[00:45:17]

Were they called 3270, those screens?

That's right. Yeah, you're spot on. I believe 3270, yeah.

IBM made a huge amount of money from deploying those screens and selling them to people. Now you move on, we're talking now just the turn into the 1990s, 1991 to '94 you are Technical Lead, Mainframes. Now, what did that mean?

Well, by that time I think there was a lot of people retiring or moving on out of that area that I'd moved into, the mainframe area, and as a result of that there were opportunities. And I had at various points thought is this, am I still in the right place, should I still be working at the Met Office or should I be using my talents elsewhere now. But I always had this affinity for meteorology and I thought it's the right thing

to do, to stay and build upon a career here. So I was offered the opportunity of becoming a team lead and that was a good thing, because managing a team of about fourteen or fifteen people, running that mainframe environment, that was a great time for me actually, yeah.

So what is the Martyn Catlow method of management?

You mean stick or carrot? Oh, it's definitely carrot, I mean I think you have to encourage people to learn and I think some of the stuff that we were involved with in that area was very challenging and new people coming in really had difficulties in understanding. Because technology's not just about reading a manual, as you know, it's about, as we were saying earlier, understanding what other people have done, and you don't start everything from scratch, you take things forward and build upon them. So I very much consider that what I did was building on the shoulders of giants, if you like, from the 1970s guys, and that's what I tried to instil into people that were coming into the team when I was running it, that you have to impart your knowledge to them in a way that makes them enthusiastic and want to continue and drive things forward. Yeah, I mean we had a lot of turnover during the nineties in the team because computing was obviously becoming much, much more predominant in the business world and there was a very high turnover of people, and people were coming, get trained up and leave. But it never stopped me taking the view that you have to be enthusiastic, you have to encourage people and you have to drive that enthusiasm in them and if they choose to leave, then, yeah, someone else has benefitted. But the UK's benefitting in general, isn't it?

This was the early nineties. You've got a fantastic set of practical skills now and techniques at your elbow. You've got your HNC, you can earn a lot more money in the private sector, why didn't you go?

I tried it a couple of times. I was offered a couple of jobs, but they never seemed to be, to me, as challenging as what the Met Office were doing, in terms of its computing and direction and where it was going. Because the Met Office, of course by this time, had its first supercomputer, in fact it was probably on its second generation supercomputer by this time, and this was really challenging for someone that both was

interested in data processing and the weather. And at the time there was, you know, commercial folks were interested in the database, management and sort of banking and that sort of thing and it didn't really appeal to me, I was still really into the weather and making sure that, you know, you got the best out of the data processing in that domain really, and that's why I didn't leave. Civil service salaries have never been top of the pile, for sure, but it's the enthusiasm of working in an organisation that's forward looking and challenging, which is why I stayed.

[00:49:43]

Now the technology that you're using has come apace, but your exploitation of it in the Met Office has also come apace. In the decade that you've been, just over a decade in the Met Office, in 1982 they were using a fifteen-level numerical weather prediction approach model. But by 1991 when you took up your technical lead in mainframes, there is a twenty-level unified model that's being processed. So the complexity of processing not only demands these larger machines, but also a more sophisticated software environment, is that right?

Yes. So – that's right – so as the models increase in complexity and the science that drives the models increases in complexity, then the processing demands of those models increase, obviously. And the time window that you've got to put a forecast out, for example, still has to be maintained. So you need more power out of the machines that you're using in order to ensure that sort of faster clock speed to get more cycles processed and more efficient code to make sure that you're using the minimum number of cycles to reduce the time of output of results. And to support that, obviously the Met Office then was looking in different areas about its supercomputing and what that could do and the move began of modelling to supercomputers while the mainframe environments were then used to do all the subsequent downstream processing. That processing was largely data processing, the supercomputers obviously doing higher performance numerical computing, and at the same time scientists who were doing the modelling, and at that time climate modelling was coming to be a very big thing, they were also getting access to dedicated machines which were becoming financially viable, shall I put it that way, and financially viable and capable for them to do their own processing at their

desktop. So things like Unix workstations were becoming available for specialised programming and visualisation of data, and they were being deployed round the office. So the mid-nineties and by the time we've reached the next millennium, the Office had changed from being an organisation focussed around batch processing into an organisation that was beginning to use interactive processing and allow scientists to make changes in real time rather than change blocks of code and resubmit them in batch for processing and what have you, to wait for the results so you could then start to do things interactively and get results straightaway. And that speeded up the development of modelling and the data processing techniques that were being applied to the data.

And you now take your praxis a step further and you get into support for and development of and exploitation of Unix operating system?

That's right, yeah. I was never a Unix expert, per se, but I came across it because we were involved as a central team in advising how – because we had the experience around the mainframe environment – we advised how these departmental systems, if you like, were being introduced and the sort of techniques that they needed to ensure that they were being used and deployed efficiently. So there was an experience, again, wherever you're ground-breaking, experience is useful, and even experience in different computer technology, if you like, is still valuable when you're applying the same principles to a new environment, like the Unix environment.

[00:54:10]

We're now approaching the turn of the century and the dreaded Y2K. You were quality assurance for Y2K preparedness, was that a daunting task?

Well, it was, because at the time, although we had the supercomputing for the modelling, much of the data output was still being done through the mainframe. So, you know, what you didn't want to go wrong at midnight 1999 was no aviation forecasts being available, otherwise that would have been catastrophic. So there was a lot of pressure and a lot of emphasis on ensuring all the systems that were in place –

not just on the mainframe, but across the Office – were ready for 2020. And as it turned out, there were very few serious implications of that change. We were lucky.

It's been put to me by some people that the Y2K was a big scare by a number of large consultancies who made an awful lot of money out of it. Is that your opinion?

I think it was overegged a little bit, and I think it was overegged, as things always tend to be, by stating to people, creating fear, uncertainty and doubt about things. And I think if you're a technician you can home in and say, well actually, the problem isn't all pervasive, it's actually some very specific areas and we can probably find out what those are quite quickly. But nevertheless, as you say, there was an opportunity for a lot of money to be made in a very short time.

Did you bring in consultants?

We did have consultants. Purely to really audit what we'd already established. And I think that's the key thing, because we're a public sector organisation we took it very seriously and made sure that, you know, for all our customers' purposes, you know, needs, that they were assured that we'd taken year 2000 seriously, which of course we had, but to do that it's good to get the rubber stamp from an independent organisation and I think that consultancy has two values. One is it helps you find out whether you've done everything properly, but also it gives that stamp of approval, an independent stamp of approval to say that you've done it correctly.

Somebody I interviewed for the Archives about Y2K, he was in charge of international IT for a private sector company, and he said, okay, New Zealand was alright, Japan was alright, but I really turned round and partied hard once Italy was okay.

[laughs]

If you see what I mean. You were just focussed on this civil service operation, which was already anyway pretty disciplined in what it did and how it did things. Next you've got another big leap, which is a physical transformation, and from Bracknell to Exeter.

Yeah, that is indeed true. That was a remarkable thing, of achievement actually, in the Met Office history. I don't think any other change so impactful had ever taken place really, because of the nature of the reliance on the Met Office by that point and its services, it was really important that the transition was made in a seamless way. And the planning that went behind that was, it was very diligent and there was a lot of people involved, a lot more challenge applied to design and concepts that were going to be used in that transition of the IT. Because if you imagine it, you've got an entire operation that's working and being established in Bracknell now for twenty-odd years, and then suddenly you're going to move that cross-country, several hundred miles, and you have to continue to produce the production outputs from the models and supply the customers with all the products that they're expecting without disruption. Yeah, that was one of the biggest challenges I think the Office has faced in its entire history.

So you have one Cray going down the road from Bracknell to Exeter and another Cray still in Bracknell working. You'd install the Cray in Exeter, get it up and going, and then unplug the Cray in Bracknell and send it down to Exeter, that was the type of transfer, was it?

Yeah. Luckily, all of the systems had resilience in Exeter and in Bracknell so that, you know, there was a capability that you could always fail over one of them, from the main production machines to another one so that maintenance could be carried out or repairs could be carried out, so that the systems would continue, the services would continue to operate regardless of whether the systems were up or down. So the principle of migrating work between one system and another was established, it's just that it had never been done over, you know, hundreds of miles across country before. In essence it sounds simple, but lining up the ducks to make sure that that all worked and particularly the communications were open and available all the time, and in those days, as I say, the fibre connections that we have today we take for granted, the infrastructure in the UK has completely changed in the past twenty years, but back in 2002 things were much more elemental and less reliable and you still had to maintain the data flow between the two sites, Exeter and Bracknell, as well as moving the equipment. And co-ordinating that was, yeah, a massive challenge for everyone.

[01:00:36]

Did you take the opportunity to junk some of your systems?

There was a lot, obviously a big clear out, if you're going to move house you're going to get rid of all the rubbish, and because it was greenfield down in Exeter we had the luxury of moving into completely new, freshly built IT halls. They were built ground up with the latest available technologies and infrastructure and some of the kit didn't migrate. There was, some of the kit was installed fresh and new in Exeter and some of the old kit was left behind in Bracknell. And that was a really good opportunity to clear out dead wood and then force a refresh of technology which was, in some cases, long overdue in Bracknell.

Did you take everybody with you or did some people decide to stay around Bracknell and Reading?

Well, obviously, I mean it's a big change for people from a domestic perspective and I think a lot of people thought that there was more job opportunity and more flexibility up in the Thames Valley, which of course there was, because by that time all of the Thames Valley had burgeoned into, as I said earlier, the Silicon Valley of the south of England, and a lot of people thought well, I don't want to move down to Exeter because there's nothing in Devon. How foolish were they? Yeah. So we did lose some good people, but a lot of people did choose to move because of the quality of life aspect.

Why did you choose to move?

Two things, I think. One, I was, I'd been in Bracknell for quite a long time and I thought it was time for a change and this was the perfect opportunity to move to an old town, an old city, much like it was up north. So that was a good move for me, because I, you know, established old part of the country, it just felt right and it felt like going home to Blackburn, but actually no, it was 300 miles south.

A bit different from Blackburn.

A bit different, but it's still an old place, you know, with an established history and so on. So that felt good. But also, you know, I hadn't lost that lust for working with meteorology and climate and that connection, by this time in my career I was thinking, that's what I'm going to stick with, really.

And when you were now established in Exeter, you launch into another blend of operating system, the Linux operating system, and you're going to lead the mainframe exploitation of Linux, yes?

Well, it actually started back in Bracknell, but yes, absolutely. It was in the last few days before we came down, around 2000. And I think we were probably very innovative, because Linux is a derivative, if you like, of Unix, but it's a much more open technology because it allows people to contribute to its development as opposed to being developed by, say, Hewlett Packard or SGI or Sun Systems, it was an open system and it was basically a place where during the late nineties, I think was the time, when open source was taking off, and open source is the concept where everyone can join in to create that code base that everyone can then share. And I mentioned earlier on where in the Met Office that concept had been used by the early developers of Fortran to create libraries that were common in use so that everyone could use them. The concept of open source was on a global scale, implementing the same principles, really, that everyone can contribute but there will become a definitive set that everyone can share of code. And, you know, that was a key change that was taking place during those days. And the advent of commercially supported Linux turned that open source drive, that community if you like, into something that businesses could start to adopt. And I'm very pleased to say that we again were pathfinding with Linux in a very early stage in the Met Office, and we implemented the first production Linux on the mainframe before anybody else had installed a Linux environment in the Met Office. So we path-found on that technology for the rest of the Office. By the time we'd moved to Exeter, it had really taken off and there were so many different uses of Linux being deployed around the entire organisation, on desktops, on departmental computers, as well as on the mainframe. And ultimately today we've got Linux running on the supercomputers as well. So the age of

proprietary operating environments and proprietary software is largely past, I think, and we're in a new world now where Linux is king.

[01:06:23]

And I would add to that – it might not be your opinion, but it's my strongly held belief, and I think I've got some evidence for it – it actually produces, this process of open software development, better code, because instead of some manager inside Microsoft saying, okay, we'd better get it out otherwise you're going to have a dip in revenue, and by the way, we know there's 120,000 bugs in it, there are hordes of geeks in the world working the open system process who are just dying to find mistakes in each other's code.

A hundred per cent. And so often during the 1980s and nineties I was frustrated by commercial software that we had to install to support, you know, facilities at the time that if something was wrong it took ages to get a fix to address the problem. If you wanted a change to the functionality it would take forever for that to happen, it was all driven by business decisions within the companies that owned the software. And now it's a completely different world, as you've alluded to, but people can contribute and they make assessments based on need rather than, you know, business need, and open source is just fantastic. But there is a dark sign on the horizon actually that's beginning to emerge into daylight, with cloud computing, where from my perspective we seem to be going back in a cycle towards platform specific code, which is beginning to become apparent on platform site Amazon where there's, you're bought in to techniques that are almost proprietary to their environment. Which is taking, in my mind it's taking us back to the 1970s where IBM was king and their operating system, System 390 and z/OS have been very proprietary. And it works against the principles of that open source in many respects. So you have to be very conscious, I think, whenever you choose to do something in computing, what the long-term consequences of it are, rather than the immediate, fulfilling an immediate need by doing something pragmatic. Always take the long-term view, make sure that you're making the right choices.

[01:09:04]

By this time, 2003 or so on, the Met Office had adopted Oracle relational databases. Why?

So, what's a good point? I think... Oracle was introduced because... we started off with a product called IBNS, which you may remember.

Yeah.

And that just, I think the fundamental difference, because that was a – I forget the terminology now, I'm not a database expert – but it was a non-relational database and I think the thing about the Met Office data was that it suited a relationship type database and that's what Oracle did really, really well. So the choice at the time I think was between Oracle, Ingres and BB2, but the choice went to Oracle because Oracle seemed to be more advanced in their geospatial capability. And by that I mean they were more aligned with the sort of geographical stuff that the Met Office did than the other database implementations, which were more oriented around business. And I think that's why Oracle managed to get a foothold and became pretty predominant. It's still predominant today, actually, we're still running much of the Oracle Database estate on the mainframe today.

Let's just move forward quite a bit, because you're now looking at next generation supercomputing, what does that look like for you?

Next generation supercomputing. Well, there's a story. [laughs] So, before this... okay, so 2020 saw a change in approach for the Met Office, because up to that point from the first machine beside the 205 which came in in 1979/80, we've always installed our own supercomputing on our own premises and managed, alongside with the supplier, that equipment. And it's been very challenging from environmental perspectives and from power supply perspectives and from working with suppliers to ensure that that capability is always matching what's needed in terms of demand from the Met Office. Scale's been a challenge. The Met Office has often wanted to increase its capability and resolution of models as we discussed earlier and the

hardware, the supercomputing hardware sometimes limited that. So, although we've run our own computing from 1983 to 2020, in 2020 a decision was made to shift it out so that it was being hosted in a different way, and that different way is basically by taking advantage of the scale that cloud providers are now employing to support their world capable data centres. And the investment that they could make in terms of both the infrastructure, power, sourcing, environmental, cooling, all the stuff that was challenging for the Met Office to do and would have been exceptionally challenging to allow the Met Office to achieve its next generation of modelling, had to move out of Met Office domain into this much more scalable cloud environment. So, rightly or wrongly, and I had different views on it, but I think my reality now is that because of the scale, because of the environmental, because of the sort of nature of our workload, we absolutely needed to make that change in 2020 to a different way of hosting supercomputing.

Were you resistant to it, to begin with?

I think, yeah. No doubt in my mind, I think my initial view of the cloud providing industry was that it was a money-making industry, because once you've got your workload on somebody else's platform you're no longer in control of how they charge for it or how they utilise it, or necessarily how much performance you get out of it. So you're handing over a huge amount of trust to organisations that are very commercially focussed. So you have to make a decision whether you want to take that risk and trade off against losing the ability to do the very best you can to get the very best out of your investment, and handing that over to someone else to do on your behalf, and for me that was a big challenge. To some degree it remains so, but I think what the Met Office is doing today in terms of its supercomputing and cloud is not cloud as general cloud is done, it's being done in a very innovative and novel way which enables us to take advantage of the scale of cloud without necessarily buying into all the dangers of cloud in terms of the economics and in terms of the lock in.

[01:15:06]

But it does take you back you thought, possibly back, you thought to a proprietary world?

It makes me think that you're potentially at risk of, you know, going onto that cycle again where you're locking in by means of being tactical in terms of your thought process rather than long term in terms of your thought process around what you're doing and your strategy. I think if you maintain... I think the fundamental thing is, if you maintain in your mind that you're working as an open source organisation and you don't lose that focus, then if the majority of what you do is still open source and can be shared and can be moved and it's not locked in to something, then the margins, five, ten per cent of what you do can get locked in and you can easily withdraw from it. In the 1970s and 1980s everything, ninety-nine per cent was locked in.

Yeah.

And it was a much more difficult proposition. Today, so long as you don't buy wholesale into what the providers are delivering in terms of their proprietary approach to service provision then, you know, you're going to be in a much better position to make choices and move around, should the economics determine that.

With your little telescope are you looking at quantum computing?

That's an interesting one, isn't it, quantum computing. It's on the horizon but it doesn't really seem to be manifesting itself in anything real. I think we're waiting, I think it'll be another twenty years before certainly the models- because, you know, computer models produce outputs of massive scale and to change them from working in a conventional architecture, computer architecture, to a completely new one not only means, you know, you have to tune and re-engineer for that architecture, but you also have to make sure the science works on that type of architecture as well. So it's an unknown for me at the moment how we would transition into quantum, to be honest.

Your data that you have in the Met Office is absolutely vital for many operations, not the least of which will be military. How do you shield yourself from cyber attacks?

Well, yeah, the Met Office has put a massive amount of effort, I mean huge compared with what we used to do back in the nineties. Since 2010 there's been a huge emphasis on cybersecurity, driven by central government and, you know, GCHQ and people like that have obviously become very much aware of, you know, the impending threats from around the world and so on. And yeah, there's been a huge investment in government and in particular in the Met Office to ensure that our services and products are not impacted by cyber threat.

Maybe I'm being very naïve about this and don't quite understand the technology, but it seems to me as the cyber threat mounts you're losing control because you're putting it up on a cloud. Isn't it more vulnerable there?

Another excellent point. And, you know, from my perspective that remains the huge risk in terms of where we are in the world and the politics that, you know, global politics and how it affects multi-national organisations and how they're potential targets for international terrorism or warfare or whatever. And again, the more you hand over your control, the more you are in the hands as to how seriously they take it, in terms of cybersecurity and installation security and all of those things that sit around computing. And it's a trust thing, again, and you have to work on the basis that these organisations have the money to make the investments that make secure IT possible at the very highest levels, and consequently they're meeting the sort of levels of expectation that you would have if you were doing it yourself. You know, I think they do, but there's always that risk because they are so big that things get forgotten or lost or processes fail and something breaks. So I think, you know, in my head, in the world in which we're in and the world in which we're heading towards, we have to be far more conscious about maintaining a capability in both environments so that you don't lose complete control of your ability to operate in the event of a major disaster. Something that takes out cloud or a network infrastructure, for example, externally, you still need a capability to be able to deliver key services to those people that need them.

[01:20:36]

Last two questions, Martyn. Question number one, what are the biggest mistakes you've made in your career and what have you learned from them?

Biggest mistakes? Oh gosh. That's an interesting one and I feel perhaps – and I'm in two minds about saying this because I really enjoy the direction that my career's taken – I think the biggest challenge has been not being able to progress through management. And what you see is, you see very talented people coming up alongside you, but because you're focussed on an engineering career, if you like, you miss out on the benefits of being able to influence on the management side and set the direction for the organisation. Even though you're still influential, you're not part of that culture, if you like. I think that might have been a mistake, but I'm still not convinced. So I'm in two minds about that, but you know, that could have been a mistake.

You're coming up for, well, now forty-three years at least in the Met, one single employer, is there ever a moment you regret that?

When I look at salary, yes. [laughs] Because, yeah, when I look at my pension. Outcome, probably yes, but in terms of fulfilment, in terms of enjoying what I've done, in terms of working with some of the best, brightest and intelligent people, and some of the nicest people that I've ever come across, actually both in industry as well, some really nice folks, I think I couldn't have chosen a better organisation to work for, actually. I'm pleased I'm still here.

And I think that's a wonderful place to end. Thank you very much for your insightful and inspiring contribution to the Archives of Information Technology. Thank you, Martyn Catlow.

Well, it's been a pleasure, Richard. Thank you very much.

[01:22:52 – ends]