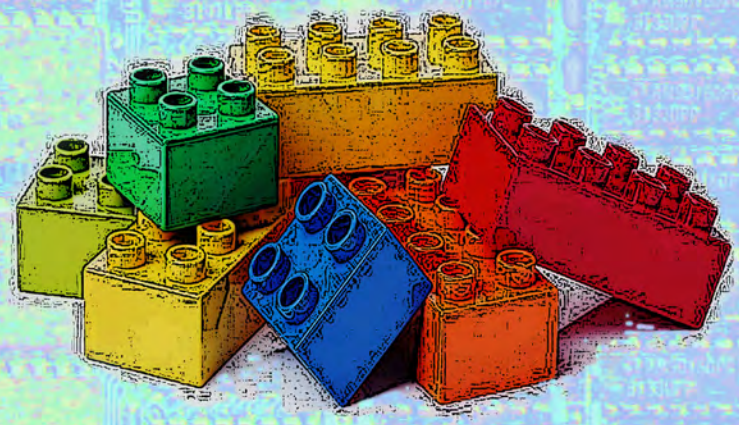
Memoirs of an IBMer

By Bob Grainger



Memoirs of an IBMer

By Bob Grainger

Contents

Introduction

I was born, Bernard John Grainger, on Wednesday, 8th March 1961. My father was Bernard Gordon Grainger, he was born 29th December, 1927. My Mother is Olive May Grainger and she was born on the 1st August, 1928. I have one sister, Stephanie, born 9th September, 1953.

My father was too young to serve in the war, but did do two years national service as a dental mechanic. For most of the rest of his career he was a Senior Mechanical Engineering Lecturer at Garretts Green Technical College in Birmingham. My mother served in the WRENs, and spent most of her career teaching needlework to primary age children.My sister taught drama at Hertfordshire University. We all four had nicknames that really stuck, namely, Bun, Ols, Steph and I am known as Bob.

We had a good sized semi-detached house in Kingsbury Road, Erdington. From there I could walk to primary school. Later I went on the bus to King Edwards Grammar School Aston, about three miles toward Birmingham City Centre. After three decent A levels in Maths, Physics and Computer Science I went to Birmingham University to study Electrical and Electronic Engineering and Computer Science.

Shortly after gaining a first at Birmingham, I married my wife Ann, who worked as a legal secretary in both Birmingham and Winchester. About nine years later we had our only son Thomas, who is known as Tom. He is now writing Computer Software, mostly web related and working in London.

After University I started working for IBM UK initially at the Hursley Laboratories, although I also worked at Millbrook and Havant sites.

These memoirs document the enormous fun I had, both working for IBM and leading up to that time.

Please be aware that I have not been entirely consistent with the use of UK spellings, for example, “disk” and “program” I have chosen the US variants, simply because, in my judgement they have taken over here. Generally though, I have tried to use UK spelling, any remaining errors are, of course, my responsibility.

Childhood technology

Some of my earliest memories are of joyous openings of huge packages of the construction toy, Lego, at Christmas and birthday times. Lego was a great favourite of mine; I think partly due to the quick deconstruct time. One day I had made a hotel, with blue roof tiles, next day I could spend 10 minutes demolishing it, ready to build a block of flats.

I could be kept entertained for hours alone, honing my creative and motor skills. I was never very interested in the models in the instruction booklets, I much preferred to do my own thing, sometimes just assembling bricks until later deciding what it was I was making. Fortunately in those days Lego consisted of a small range of bricks, so encouraging imaginative play.

I remember one occasion when I was taken to a toy shop in Six Ways Erdington by my Nan, Lizzie. She, my father's mother, was born in 1900, I remember because she explained to me that working out how old she was, was easy. Our mission was to supplement my Lego collection with a few small boxes of extra bricks, since inevitably the kits were always lacking a few bricks of certain colours and types. She told me that there was no rush, so of course, I spent about an hour going through the inventory, choosing the best set of three small boxes of extra bricks, while she patiently awaited my decision.

Her husband, who died when I was about two, was one of many that exaggerated his age to serve at age 17 in WW1. He spent time in the trenches, but only the last few months of the war, so was one of the lucky ones to return.

At that early age, I suppose about five years old, so 1966, my favourite book was a hardback children's history of mathematics. Just as well it was a hardback, since the use I made of it, a paperback would have long disintegrated. The book was full of colour illustrations, of Babylonian stone tablets with their strange numeric markings for accounts. There were also pictures of Greeks doing their geometry, and the five platonic solids. I knew every page and every word, and would ask anyone to tell me a story based on a page of my choice, hoping to discover some new unwritten nuance.

Dad made a blackboard, that was mounted behind the kitchen door. The first thing he wrote on it with chalks, were my times tables. He explained to me that for each table that I could recite from memory, he would give me a tanner or 6d, which was six old pence before decimalisation. When I could do them all up to and including 12 to his satisfaction, I could expect a ten-shilling note. I believe that learning tables early is key to later success in maths.

Next bit of technology I remember were the polystyrene cutters. Polystyrene ceiling tiles were all the rage, evidently no-one had realised the fire risk. Anyway, Dad made these mains powered units, using a wooden box about the size of a large shoe box, including an upright, supported piece of wire fire element. A mains transformer inside the box powered the fire element, which glowed red hot. Pieces of polystyrene tile or spare packaging material could be cut in lines or curves, by dragging them past the wire, taking care not to inhale the fumes or touch the wire! Health and Safety clearly wasn't of interest to me or anyone then.

I took mine to school, and Dad had to make three more for the art class, they were made from spare parts so all looked a bit different, but they all worked fine. The highlight for me was when I made a model of the much talked about lunar lander. I modelled it from a large poster, and the final result looked really realistic, apart from being white. I had a can of silver spray paint, but my teacher cautioned me not to use it, as it might melt the polystyrene. He suggested spraying a small scrap piece of polystyrene, to see if was OK. Well I did this, and after a few minutes it seemed fine so I sprayed my lunar lander silver. This improved it no end, it looked really great, for about 20 minutes.

Of course the lander (along with the scrap piece) melted into a white and silver mush. Oh dear – I hadn't waited long enough.

My sister, Steph, seven years my senior, had a close friend, Sue, they were about the same age and pretty much inseparable. Sue lived next door but one to us, but they had each been running up a bit of a phone bill, nattering for hours when they were only about thirty feet apart. Solution to this was interesting, since they both occupied back bedrooms , they hooked up two tin cans on a string, and strung it between gardens. To ring the phone, they just rattled the string, which produced a clattering at the window at the opposite end. The only slight problem was that our immediate neighbours couldn't open their windows properly, because of a taut string passing by, next to their window panes.

A few years on, after Steph left for college, I became interested in Air-fix models starting with the classic Spitfire. I was somewhat disappointed that flight was impossible with these small plastic models. When a Keil Kraft balsa model arrived one birthday, the prospect of a proper flight seemed much more exciting. The first models I built were either gliders or powered with twisted rubber bands. Once again I was disappointed, the flights were uncontrolled, and over in seconds. I wanted long and powered flights. The solution became a new hobby and was called control line modelling.

The planes were powered by small glow-plug or diesel engines, and again balsa wooden frames with tissue tightened and strengthened with polyurethane dope. They were controlled using a pair of wires probably 15

yards long, which went from the wing tip to a control handle that could be used to control the height of the plane, via a small triangular crank connected to the elevator on the tailplane. There was some amount of learning involved in flying these models, so the starter kits were called trainers and were relatively slow and not too responsive. I soon got the hang of flying these, but whilst five minutes in the air was a vast improvement, I soon became disappointed with these since they were too slow compared to the looping, flying wings piloted by experts at the park where I flew.

Whilst I built many control line kits, gradually increasing in power and speed, the vast majority got wrecked in crashes. Each time I destroyed a plane, I lost what would have been several weeks spare time work, although the engines generally survived. Dad called them “brrm-clunk” models, referring to the sound of a crash, and the previous short flight time! As a hobby it was nowhere near as expensive as radio controlled planes, and the instant feedback from the wires made it feel really like piloting a plane.

I didn’t get disheartened, I just kept wanting bigger and faster models. One birthday, I received a beautifully made, large Italian engine called a Super Tiger, these were the Lamborghini of the model aircraft engine world. I hunted around for a suitable kit to build it into, but they were all too expensive. So instead I found a small plan in a book, for a flying wing, which I scaled up and built by hand to accommodate my new engine. The engine mount was particularly tricky, since this had to be a custom fit and the plans had no details as to how to arrange it. Rather than using tissue, I covered the plane in a shiny black plastic-like substance called Solarfilm. This could be stretched to fit the wing frame using a warm air setting on a hair dryer.

Well flight day came, we took it to the park, and I asked one of the experts there to pilot it for the first time, since I needed to be gently introduced to flying it. The chap’s name was Ken, he must have been in his thirties or forties. He looked it over and was pretty impressed, double checking again that I was sure I wanted him to trial it. We turned over the propeller a few times, but the engine wasn’t interested in starting. Ken connected his electric starter, then it sprung into life. He tuned the throttle down to starve the engine a little and the screaming noise was like nothing we had heard at the park before. The power from the propeller was trying to rip the wing out of my hands, so Ken trotted to pick up the control handle, I tensioned the control lines, he gave the thumbs up, and I launched it outwards and upwards. The next few minutes were a sight to behold, once off the leash this machine seemed to scream even louder, and flew like a bat out of hell. Ken was perfectly at home flying it, he had it looping repeatedly, and figures of eights, etc., it would clearly hold its own in any dog fight, which was possible with these models, using a trailing ribbon as “chewing material” for the opposition’s propeller. The screaming engine sound was modulated by the looping, (the Doppler Effect) so Ken seemed to be playing it like a lead guitarist.

We watched in admiration, as Ken was clearly having a lot of fun, then I started thinking it would be my turn next, and my heart was in my mouth since I wasn't sure I would be able to fly it, but what the heck I was going to have to try. Well sadly that was never to be, suddenly there was a click from the elevator and despite Ken's best efforts to minimise the damage, the plane hurtled at full speed from full height vertically down into the ground, easily 60mph, bouncing about six feet. You would think that the spectators would have laughed at this calamity, but no, everybody in the vicinity was disappointed and sad for me.

We all rushed over to the wreckage, and discovered that the piano wire link to the elevator had not been secured well enough, which was of course, my fault and the huge pressures had unhooked it, all it needed was another 90 degree bend and the accident couldn't have happened! The aircraft was a total write off, and the engine wouldn't crank at all, since the crankshaft must have become bent on impact.

Dad managed to straighten the crankshaft using an instrument called a dial test indicator, along with a hammer and a bucket load of skill. We proved that it still ran, but exams and other interests got in the way of control line modelling, and that was my last time at the park. Even though I had no time flying this model, finally I wasn't disappointed, this build, although short lived, had been an unforgettable success.

Next thing I made was a matchbox radio. This used a three pin integrated circuit called a ZN414, which although obsolete, still has documentation available on the internet today. The box was a bit bigger than a real matchbox, but was made of balsa wood, covered in some left over black Solarfilm, leftover from you know what! The sound quality was awful but it was great in boring lessons, at least until I had it confiscated.

Another project I remember was a telephone ringer simulator. It was designed for use in the theatre, intended to indicate to the audience, what an actor making a phone call, would be hearing. The sound was much as you would expect, i.e. "ring-ring pause ring-ring pause, etc.". I've absolutely no idea why I wanted to build this circuit, I had no intention of using it in a theatre. It was the most complex circuit I had built, with about fifty individual components. I made a wooden box about half the size of a tissue box, and found a use for it in pranks at school. The faint sound of a phone ringing in a crystal earpiece is surprisingly distracting, everyone wants to know where it is coming from, in case it needs answering! This unit also got confiscated.

Dad made a metal detector, this made an irritating whining sound, into headphones, that squealed when sweeping over metal, there was a whole load of large coil winding involved with this, and he got it working down to about eight inches depending on the size and type of metal. He had built it

specifically to find his Mother's gold wedding ring, which he had inherited from her a few years earlier, and promptly lost whilst gardening. We spent ages scouring our garden in Erdington, but found only rusty nails and the odd boring dirty coin. Dad did find his Mom's ring, but ironically enough, only when digging the garden to plant the veggies.

There were other projects, some such as power supplies, audio amplifiers, oscillators, strobe lights and digital sequencers for discos and several long forgotten other things. Looking back I'm amazed I had the time considering school pressures. The projects were predominantly made with spare parts, either from old tellies or previous projects. It was always a bit of a triumph to make a project without sending for any components, since this saved money and time, since most parts were only available through slow mail order. We were more than happy to try substituting parts to avoid waiting and paying. Keeping the old tellies was always a battle between Dad and Mom, as she wanted to get rid of them to tidy up. Sometimes Dad and I had to de-solder all the useful components, each of which went into a carefully indexed metal shelving drawer system, just so we could dispense with only the empty TV carcasses, to placate Mom.

The interesting thing about all these hobbies: Lego, model planes and electronics, is that they all shared this process of create, use, scrap and recycle sequence. It may have come from the make do and mend initiative during WW2 (1943), but I think Dad's recycling instincts predate that since he made and sold wooden push along toys for toddlers before the war.

Each of my hobbies was training for the next, and of course for the rest of my career, but before we get to that the next chapter relates to some adventures at school.

Online Dating in the 1970s

Third year at school for me was an exciting time, I was 14 years old and had a term at Longdon Hall (a country house and farm, with dormitories for about thirty boys) to look forward to in the spring. However, it was the autumn term and for one lesson a week we had the intriguing prospect of a new subject entitled, "Computer Studies".

Well the reality turned out, initially to be rather drab, we were taught the rudiments of BASIC programming, and our homework was to fill in a pro forma grid with our attempts at a program. We had to hand these in the next day, they were parcelled up and sent off somewhere, I knew not where. These grids had to be filled out very accurately, so we had to ensure that all zeros

were crossed, and O's were not, similarly ones and ells had to be very distinct. There were anonymous data entry girls that were trained to transcribe exactly what had been entered into the grid, which inevitably wasn't completely correct. The result being just a miserable error message the following week. We soon came to despair at ever getting anything to work.

Well one week things were a bit more interesting, the teacher, actually he was our maths teacher, had set up a teletype in the classroom, with a dangling long telephone extension cable to a telephone socket down the corridor. He also had an acoustic coupler connected to the teletype. He picked up the telephone receiver, dialled the number for the Birmingham City Treasurer's mainframe computer and we all waited. We could clearly hear some chirping tones, that were obviously not a person. The teacher plugged the phone's receiver into the acoustic coupler, which quelled the tones.

Then as if by magic the teletype started clattering, and he entered the login account and password, explaining to everyone what was happening as he went along. He then ran a lunar lander simulation program called LUNAR.BAS (the .BAS signifies a BASIC program) and took suggestions for the thrust values as we attempted to "land" on the moon, which was, of course, big news around that time. Next we ran GOLF.BAS and had to supply club suggestions, and swing strengths to complete a round of golf.

I was completely captivated, this was infinitely more fun, than the one week pro forma things, and I did everything I could to get more time using the teletypes interactively. They were available before school, break times, lunchtimes and after school.

As far as I was concerned, from that point on school consisted of time playing on the computer, interrupted by annoying lessons. There were only four teletypes and we only had one login account. So three of the teletypes could only be used to prepare BASIC programs, by typing them out onto paper tape. Whilst the remaining teletype could be used to run programs interactively. There was a continual queue, which even degenerated into fights, when misunderstandings arose about whose turn was next.

Each of the four teletypes had acoustic couplers and phone lines, so I started wondering if there was some way that we could use more than one at once. All we needed was three more account numbers and passwords. Our account number was a four digit number, which I noticed corresponded to the last four digits of the school phone number. Our password was "KEGS" which stood for King Edwards Grammar School. I started to wonder if the same scheme was in use at other schools. So I looked up several schools in the phone book, and made guesses at their account and password. For example, Marsh Hill, I tried "MARS". Worked first time. With help from friends I was able to guess about fifty account/password combinations this way.

Sometimes I found that the account was already logged in, presumably, the legitimate school kids from that school were using their own account at that time. So I kept a careful record of which schools were using their accounts the least, and we stuck to using those accounts to minimise inconvenience to other schools, and also, of course, to avoid getting caught. We found several accounts that seemed to never be logged in, so we used those to keep out of trouble. All four teletypes were then in continual interactive use to run programs. The phone bills would have quadrupled, but since that was paid by the school we weren't bothered about that.

So the world was our oyster, we had access pretty much whenever we wanted, and could also nose around at files created by other schools. This led me to a new idea to meet girls. What I needed was a way to get the girls to run a program, this seemed to be impossible, why would they run a program other than one of their own I needed something that would be eye catching in the directory listing. So I called the program SEXY.BAS, and set about ensnaring those girls! The first thing the program did was to ask "HELLO, ARE YOU BOYS OR GIRLS? (ENTER B/G)". If they answered B, the program deleted itself from the directory, (we didn't want any male competition!) and terminated.

If on the other hand they entered G, for girls, the program said, "HELLO GIRL NUMBER 1, PLEASE ENTER YOUR FIRST NAME". If they entered say "JANE", the program replied politely, " HI JANE, PLEASE ENTER YOUR HOBBIES OR INTERESTS". Then it requested hair colour and a phone number, and a time that would be convenient to call, saved the information in a data file, and looped around to girl number 2, etc.

We produced four paper tape copies of the SEXY.BAS program, and transferred them to as many schools' accounts as we could log into.

There were no lowercase letters in those days, so everything was in capitals. There was also no way to transmit the girls' names and numbers in the data file to us, so we had to manually log into each school after a few days and check to see if there was a data file present.

Well the program was found and run by about two or three schools, one was a boys school, and they rewrote the program so that boys were catered for, obviously we ignored them!

One school seemed to have created a data file with three girls in, so after school I tentatively dialled one of the numbers, and talked her and her two friends into an afternoon at the Silver Blades ice rink in the city centre. Amazingly enough all three of them showed up, along with me and two lads from KEGS. We had a reasonable time, apart from the fact that it was their first time on ice, so they spent most of the time clinging to the side rails. I

remember being dressed in a fetching royal blue blazer, but towards the end of the session, falling into a puddle of water, and getting soaked through, so as well as looking like a complete numpty, I had to endure the bus ride home with a wet backside. The girls were from the opposite side of Birmingham, and clearly not interested in a repeat skate, so no lasting relationships transpired.

When we turned up for the next Computer Studies lesson, there was somebody from the City Treasurer's department, and a sombre looking maths teacher. We had been caught, and we all had to write letters of apology, although, I always had the impression that our maths teacher was secretly a bit proud of our ingenuity. I'm not sure how they tracked us down, probably another teacher had found a copy of SEXY.BAS, but there wasn't anything to indicate which school had installed it. Our maths teacher certainly didn't block our access using the four teletypes, and the other schools' passwords. We just deleted all copies of SEXY.BAS and carried on having fun. Perhaps the extent of our deviousness wasn't entirely clear to the authorities!

Anyway, that was the end of my forays into online dating.

Our first real personal computer

I've some troubles pinning down the exact year of this story, but I was definitely at school, somewhere between fifth year, and lower sixth, so aged about 17, around 1978. I had used computers at school, but I knew having our own machine would be a big step forward in terms of learning and access whenever wanted.

At the time, the were few choices, there was a kit machine called a Nascom, this was an open board machine meaning it didn't have a box or even a power supply. It was intended to connect to a TV screen which also had to be provided. This machine was about £200, was selling well and was almost ideal, apart from the fact that I disliked the Zilog Z80 microprocessor it was based on. The instructions were somewhat messy, and it wasn't compatible with the Motorola M6800 we used at school.

Clive Sinclair had a very cheap but limited machine called an MK14, which I ruled out as useless and uninteresting. It had a few LEDs (light emitting diodes) and a horrible "chiclet" keyboard, which only had about 20 keys.

So I scoured the electronics and computer magazines looking for an alternative. In the small ads I came across a company called Hewart Electronics, and again for about £200 they were offering an open board machine, with a video display capable of 64 characters in 16 rows, a full

alphabetic keyboard and over 1000 characters of user memory for programs. Also included was a cassette interface for saving programs. All very similar to the Nascom, but this machine was based on my favoured M6800 microprocessor.

Well £200 in 1977, is equivalent to about £1,100 in today's (2017 at time of writing) money, this was a big deal for a schoolboy. So I had to persuade Mom and Dad that this was worth the outlay. Dad was curious about computers, and had a wealth of electronic experience, but hadn't programmed before. He was reasonably keen on the idea but didn't want it to be an expensive "one hit wonder", which Meccano had been for me. What he suggested was that I help him paint the house external woodwork, and when complete, he would cough up the money. I readily agreed. This was our plan for the summer holiday.

The house painting (a fetching shade of apple green) was a complete pain, a lot of the woodwork was rotten, so we had loads of work preparing and filling before the more fun part of painting could start. Also for upstairs, Dad didn't want me falling off the ladder, so my role was to stand at the bottom, ensuring it didn't slip and passing up and down tools. This was unbelievably tedious, made tolerable only by having Radio One blaring away, I remember one tune called "Gordon is a Moron" by a daft bloke that called himself Jilted John. It was quite catchy, if a little much after the fiftieth hearing.

Well this torture went on for about three or four weeks, I'm not sure if every last bit was done, but most of it was and it was good enough for another five or ten years.

I wrote off to Hewart Electronics, and Dad furnished the cheque. In those days companies could delay 28 days without issue, so I knew I had to be patient, anyway we went off on our holidays, probably camping to the Gower Peninsula, a family favourite.

Not long after we got back a large parcel arrived, with the circuit board and all the chips, it even had a cassette tape, with "Grainger Test Tape" hand written on it. Only slight downer was a note saying – "keyboard to follow".

The whole thing did look a bit amateurish, as if it had been made in a garage. The circuit board was single sided, so there were hundreds of wire jumpers that would have to be soldered to the top side of the board, including about a yard of rainbow ribbon cable. Nevertheless, all parts but the keyboard were sent. The instructions were printed out using what I think was called a Gestetner machine, a kind of spirit duplicator as photocopiers had not become common back then. Quite a lot of it was hand written and hand drawn.

The circuit board was about 12 by 12 inches, of which a quarter, so a rectangle 3 by 12 inches was the video display driver, including what was called a UHF modulator (UHF=Ultra High Frequency) this converted the picture into a form suitable for plugging into an aerial socket at the back of a TV. The instructions suggested constructing this section first since it could be tested independently of the rest of the system. There were about 50 chips that comprised the video display including six small memory chips to hold the characters being displayed, and one larger chip to hold the font, i.e. the shape of each character.

I spent the last few days of the school holidays, soldering all the chips for this video display. I carefully checked everything, particularly for shorts. Finally I plucked up courage to power it on and hook it up to our TV using an old aerial cable we had. The picture was mostly fuzz, but there was obviously something working, since the fuzz changed when I unplugged the cable. All it needed was to tune in a channel to match the frequency of the modulator, and bingo: a stable picture of 64 by 16 rows of random characters. Why random? Answer was that without the rest of the computer built, the display memory wasn't set to anything. So display working fine.

Well construction of the rest of the machine continued, just at the time school starts again, this was a bit of an interruption. Anyway probably three weeks later I had completed assembly of the whole board, including all the chips, along with the expensive microprocessor itself. One nice feature that Mr Hewart had included was a LED test probe that could be used to check for high/low signals and for pulses.

The keyboard still hadn't arrived, so I put off more testing. What we did at this point was to make a box, and build a separate power supply. Dad designed the wooden box using the case of an old TV, but placed flat. He added plywood base and removable top, along with sections inside to support the power supply, circuit board and at the front, the keyboard.

Gradually it was all taking shape, when one day the keyboard arrived. This required some carpentry and extra ribbon cable to wire in, but finally we couldn't think of any excuse not to turn it on. We had it in my room, with an old black and white TV hooked up next to it. There were also cables to a Philips cassette recorder on my bench.

Well we had a final check of the instructions, with Dad asking all sorts of questions as to if I'd done this and that. Eventually I just hit the power, and fuzz again, damn we needed to tune in this TV too. Once we had done that, the screen was obviously in sync, but completely blank. Hmm.

We powered it off and on again, this time, the random characters appeared for a split second, and we could see them cleared from top to bottom and that was it. We hammered away on the keyboard – no effect whatsoever. Total disappointment. Well it was getting late and Dad told me to sleep on it.

Well for several days I battled with this machine, and we both went over and over the instructions, checking all the chips were in the correct places, basically we checked every single part and every single pin. We even tried playing the test tape into the blank machine, but in my heart I knew nothing was happening.

The instructions included a listing of the software, this was stored in an EPROM, it was supposedly “MIKBUG” compatible, which I had heard of from school and sounded like it would help be more compatible with the school machines. I looked through this software, and reasoned what it would be doing at the instant of power on. The answer was that it called a subroutine to clear the display memory, returned from that subroutine, and displayed a single asterisk at top left as a simple prompt. The problem was that the asterisk, just refused to appear.

Well Dad started asking about jumper wires, and I just complained that we had already checked and double checked them. He was moaning about one particular jumper wire that was shown in the instructions with a dot in the middle. I looked again and noticed this was near the stack memory chip used to store return addresses for subroutines. This started to click into place perhaps the return from the clear screen subroutine wasn’t working. We took the ply lid off the machine, and inspected the place where this jumper wire with a dot was. Sure enough, my jumper wire was covering a small hole next to the write pin on the stack memory. So what Mr Hewart meant by this dot was a DOUBLE jumper wire linking the three points. Nothing had been allowing writes to stack memory, therefore it cleared the screen, attempted to return back to display the asterisk prompt, but went off to some random address and then looped around forever. We later referred to just as jumping off into the wild blue yonder.

So we carefully rewired this single jumper into a double jumper, put the machine back together, powered up and bingo, the asterisk appeared, and typing on the keyboard yielded the characters being typed on the screen. We knew instantly, that it was working, and Dad congratulated me, and we vigorously shook hands. A moment etched in my memory. I tried some commands to display memory, and to edit it, everything was working fine. We tried the test cassette tape, and after some fiddling with the volume controls on the cassette drive, it displayed something like “Hello Mr Grainger, Welcome to Hewart Microelectronics, congratulations on getting your system up and running”.

The machine had no further bugs. The only thing that ever went wrong related to the washing machine which at various points in the cycle, would glitch the mains. From my room sat at the computer you could hear it start the spin, there would be a flicker on the screen, and sometimes memory would get corrupted which was very annoying, we tried adding mains filters which helped a bit, but I don't think we ever eradicated the problem completely. I tended to let Dad use it when the washing machine was on!

Well was this machine a one hit wonder? No it certainly was not. I wrote hundreds of programs for it. I hooked up all sorts of things to it including an old Creed teletype that looked like it came from the war. I built a prototype of my speech recognition system for university, and even sold a 3D noughts and crosses program for 50p, making £3 in all! I learnt my trade using that machine. Dad also wrote hundreds of programs and got to understand the workings of computers, and the art of programming. The fundamentals of the execution of a program have not changed in any substantial way since the first stored program machine run on June 21st, 1948: Fetch, execute. Which means fetch an instruction, execute it, repeat.

In later years we both built and bought newer machines. A machine today is easily 1000 times faster, and has a million times more storage but somehow, none had quite the appeal and excitement of our original Hewart machine.

Ding dong!

Part of my early experience in electronics involved installing a home made doorbell. The idea was to use a copper touch plate, separated into two halves, which visitors were supposed to touch at the boundary, at which time a tone was emitted from a speaker on the back of our front door. I installed it in the summer, and it worked pretty well. However, come winter time we had some damp, foggy days, unfortunately, this wetness got onto the touch plate, and although it didn't set the tone off, it did start making an annoying faint but continuous droning sound. The dog, (pooch) didn't like this and she trotted downstairs from her unofficial guard space at the top of the stairs and started howling and growling at the front door as if someone was there.

I could dry the touch plate with a tissue, but that didn't fix it for long, so only way to shut it up was to take out the batteries, and hope visitors realised to bang the door instead. All very disappointing, so when I chanced upon an advert for a musical door bell kit, called a Chroma-Chime in an electronics magazine for about £15, it seemed a nice solution, although it used an

ordinary bell push switch rather that being touch sensitive. The company called themselves Chromatronics.

Well the kit duly arrived, and consisted of one integrated circuit and about thirty other discrete components, including two tune selection switches, from which you could choose 24 different ditties. The unit included a small speaker and I think two PP3 battery clips all housed in a white box about 8 by 6 by 2 (inches). It took me about two hours to assemble, and worked just fine. I installed it on the front door, and great, we were the talk of the town, some people visiting just to hear our new doorbell! Pooch was happy and soon got used to the new sounds of people arriving at our house.

Well my uncle John visited one day, and of course he had to have one too. I think I sent for another for him and gave him the assembled unit. Anyway, he was an experienced, customer facing BT engineer so he installed it at his home no bother, and I think he had his working from back as well as front doors.

Next time John came visiting, he had an idea for a new business, he reckoned he could flog these doorbells for £25 and make a nice little profit. This sounded great to me and so I wrote off to the company, and asked for bulk buy prices. I think we could get them down to about a tenner if we bought about twenty units. I reckoned I would be able to assemble the kits at about a rate of one per hour, if I got organised and assembled a few at a time, in parallel. John was happy to do the sales pitch, and he even created a briefcase with a demo unit installed inside, ready for customers to try. He would also do the installation including wiring and bell push switch. He could mention the units when attending to BT calls, so there were plenty of opportunities for sales.

Well this business soon took off, even my mother-in-law, Eileen shelled out twenty-five quid for one in their veranda, since all their guests came through that way via the back door.

In the end I think we made and sold about fifty units, but the profits were small: batteries, bell pushes, wiring and clips, plus travel costs if John had to attend to any customer complaints, all cost money.

The final nail in the coffin was when some idiotic customer complained to BT, which resulted in all sorts of confusion. BT were clearly not happy about servicing someone else's doorbells! So we decided to call it a day, and bank our profits accrued up to then. I don't remember any unsold units so I don't think we were out of pocket. "Don't worry Rodney. This time next year we'll be millionaires."

I think this was around 1980, so I would have been part way through my course at Birmingham University. At the time UK technology companies were paying an annual salary of around £6000. I remember thinking that this works out to about £20 for each day at work. So to start a business selling bells, self employed, would require us to construct, sell and fit at least a handful of bells EVERY day, clearly an unlikely target. I wondered how on earth companies like IBM, ICL, Motorola, BT could afford to pay graduates, most of which, start off pretty useless, so much.

Anyway lesson I learnt here was that, unlike Bill Gates, who had famously dropped out of his degree in 1974 to start Microsoft, I should continue with my studies, get decent results and apply to one of those wealthy technology companies.

The kits must have been pretty robust, John is still using his!

Interview Technique

Third and final year of my degree was a busy time, I had to revise for my final exams, complete my project and accompanying dissertation, plus apply for jobs and attend interviews.

The interviews were generally fun, some were held on the university campus, and for some odd reason were known as the milk round. Most of the big technology companies were involved in one way or another. I applied to about half a dozen of them, I think the first was Westinghouse Brake and Signal, and it wasn't long before they made me a reasonable offer. I soon realised that it is a little hard to play the game, because you can't hold on to one offer indefinitely whilst trying for others.

Next one was a small company call MicroFocus, they were based in London so I got the train to Euston station, hailed a taxi, and gave the driver the name of the tiny cul-de-sac somewhere in the middle of London. I deliberately didn't help him out with the district or more details, since I wanted to test if the "knowledge" was really true. He didn't bat an eyelid and we were there in ten minutes despite heavy traffic. When I arrived the structure of the company was made very clear to me. One guy in charge and everybody else. Total employees were about ten. The chap in charge was quite intimidating, he had a sharp haircut , and flashy cuff links. He told me that if I worked hard and succeeded, I could expect a 25% pay rise. I was thinking well that still wouldn't help much with the cost of property in London. I didn't like the guy, he seemed to thrive on stress and confrontation. Nevertheless, I still got an offer, but turned it down.

Philips had arranged an interview at their site. The odd thing was that the date was a bank holiday Monday. This worked out quite well since it didn't clash with anything at University. Well the day came and I hopped on a train for wherever in middle England their site was. After a taxi ride, I was dropped at the doorstep of a quite isolated, tall office block. I approached the entrance, but greeted by a locked door. I got out my interview letter and checked the details, I certainly had the right date and time, and the name of the block matched what was printed in the letter.

Well I hung around for a few minutes, then another taxi arrived and dropped off some other poor interviewee. I explained to him that there didn't seem to be anyone to greet us, and we compared letters and his was much the same as mine.

He tried the same locked door, funny how people always do that, I told him no-one had been in or out of there in the last 10 minutes. The entire building looked empty to me. I suggested , rather optimistically, that this might be the pick-up point and we should give them 15 minutes to show up.

Conversation between the two of us dried up, and the weather began to turn blowy, it seemed like we were in some weird lost world movie set. Finally, the 15 minutes elapsed, and still no one arrived. I was about ready to call it a day and head home.

He got his letter out and suggested calling the phone numbers on there, there was a call box not far so we walked to that and he called, eventually one dropped through to a security officer somewhere, who said he would phone around and get back to us. Well he finally did and told us there had been a mistake, and someone was on their way. A further half an hour later by which time we were both frozen stiff, a car drew up, and picked us up. The guy apologised profusely and said that the bank holiday should not have been used, and most people had been contacted to rearrange, but we must have been missed.

He said he could either drop us back to the station to return home, or at the hotel where some other interviewees were down from Scotland, for their interview the following day. I decided to press on to the hotel, even though I had no overnight bag, so had no toothbrush, and had to reuse my underpants, etc. What a fiasco!

I met a future colleague, called Gordon, at the hotel, (Gordon appears in a future chapter about pea soup!) We both got offers from Philips, the next day but both turned them down!

I had also applied to IBM UK, but their interview date clashed with my university project inspection. I weighed up the pros and cons and decided that my project was more important, so I called IBM and asked for an alternative

day, unfortunately, my day was the last scheduled date for interviews, so we were at an impasse and we reluctantly agreed to abandon the interview for me.

Well about two weeks later, I was in the middle of revision, when a welcome distraction of a phone call rang out. It was my fellow student and friend Pete. It turned out that IBM had rearranged his interview, and he very kindly pointed out to IBM that I was in a similar position, and so wangled me an interview on this new day. To cap it off he offered me a lift down to the Hursley Labs where the interview was to be held. Well this was great news and I thanked him and we arranged a meet-up time at my house.

IBM Interview day arrived and we headed down the A34, in Pete's little yellow mini traveller van, me navigating, he driving. Took us about a couple of hours. We arrived at the site and were immediately impressed. It had a multitude of office blocks and a very picturesque country house. We parked and struggled to find the correct reception. With some help from IBMers we found the correct entrance and were taken promptly to a large office which would be our base for the day.

There were about twenty applicants on that day and we were told that this year IBM labs was expecting to take on about 20 hires, from a total of about 80 applications. They usually make a few more offers, than places, since not all are accepted. So probably about 1 in 3 chance of an offer. We were the last group, and the numbers had been extended, which checked out with the curious arrangements for me.

There were three separate interviews, one personnel, two technical. Then an aptitude test, and lunch with recent hires where we could ask "anything". Finally a roundup and then we were free to go. They started with coffees and a brief presentation of the work being done at Hursley. This included displays, storage, and CICS which is the customer information and control system, that handles pretty much every electronic transaction worldwide and still does.

The aptitude test wasn't a lot harder than O level maths, mostly quadratic and simultaneous equations and looked like it originated from the 1930s, my exam paper copy had pencilled scrawls on presumably from some previous, strange applicant. I completed it and handed it in.

I remember one of the technical interviews, with a programming manager from displays. He asked about my speech recognition project, he was interested that it used the Motorola M6800 microprocessor, since a recent display project of his, had used the very same chip.

I drew a block diagram of my recogniser, and explained how the inner software loop that captured four formant frequencies in parallel. A "formant" describes a band a frequency that characterises a vowel sound. He clearly understood

my lengthy description, which was aided by his posh white board and strong smelling, fancy coloured pens.

I complimented him, saying that he was the first person to really understand my project properly, which although true, was a bit of a grovel. However it drew a broad smile and I felt like I'd won him over. The only stupid thing I said was that I wanted to do both hardware and software, which ruled me out of his own department, which hindsight tells me would have been fine and certainly better than where I started.

The personnel interview couldn't have started better, the lady seemed to be bursting to tell me how well I'd done on their daft aptitude test. I started to wonder what the other candidates were doing, since as mostly science undergraduates in the middle of revision, they really ought to be able to solve a few basic maths questions. Perhaps they had taken notice of the pencilled scribblings on theirs.

She asked me about occasional shift and overtime working, which didn't sound unreasonable, and wasn't. She also asked about travelling abroad which I agreed to, as at this point I hadn't become flight phobic. She asked if I could touch type, and how fast. I told her that honestly no, I couldn't, although I'd grown up using teletypes, so was reasonably comfortable, although nowhere near the 120WPM my soon to be wife, Ann could achieve. She asked if I would object to attending a typing course. I grimaced and asked her how long for, and she smiled and said it would just be an hour a day for the first month of work. I thought about this and we agreed that it would be a small price to pay if my future career was going to be in front of keyboards all day.

Lunch in the cafeteria was fine, then we were shown around Hursley House, and the library. The whole place had the feeling of a posh University.

The roundup at the end didn't add much, although they arranged to pay our expenses that day, to avoid us being out of pocket. Pete worked out his mileage, but of course for me there was no claim. However, the secretary dealing with the finances, pressed a crisp, new £5 note into my hand saying no matter, buy yourself a sandwich and a paper for the return journey. Really nice touch, that I remember to this day.

Well, sadly for Pete, he had a letter of rejection a few days later, whereas I had a big package of stuff pertinent to my IBM offer. I've always felt guilty about this, but there really was nothing I could do about it.

IBM were offering a starting salary of £6500 to undergraduates, and only a few hundred more to Ph. D's making me kind of pleased to be starting work rather than another three years education. The offer from Philips was lower, as was Westinghouse and one or two also-rans. It was clear where I was headed.

Moral of the story: Make a decent job of the interview process if you want to attract decent staff.

Early IBM days

Starting work for me was all a bit of a rush. There were two starting options, early July or September. Ann and I had just married on July 17th, 1982 and with a honeymoon in Ostend, there was no chance to make the early July date, but I was keen to get started, so I arranged to start July 26th instead. IBM has a policy of hiring and firing on the sixth of the month, so I'm not sure how that happened.

On my first day it became apparent that few colleagues had chosen the July date, so I started with very few people as candidate friends. My decision to request working on hardware and software, landed me in a strange department that supported very old vector display screens used for a very early form of CAD (Computer Aided Design). A vector display is one that forms a picture with individually drawn lines, unlike the raster scanned displays in TVs. The screens looked a bit like large oscilloscopes, but without the dials and knobs. They each came with an accompanying light pen, which could be pressed to the screen to select points and lines, etc. When engineers were working on these screens, there was a continual rhythmical ping-ping noise caused by the light pen clanging against its screen. The screens were known as IBM 3250s.

Huge companies like Boeing designed their aircraft using these screens. On the one hand this was space age technology, but on the other it looked like something from the fifties.

I was shown around the lab by a friendly customer engineer, called Dick. It all seemed quite fun and he gave me a manual so that I could play around with some of the screens in the lab. The lab was of the raised floor type, and for some reason a few floor tiles were up, probably for maintenance. Anyway, whilst I was chatting to Dick, I stepped back, but my foot met no resistance, and like a slow dream I started to fall backwards into the deep hole (about three feet deep). Luckily Dick was on the ball and grabbed my arms preventing me from a serious injury. I was in a bit of a panic, but he just laughed saying, "We nearly lost you there, Bob". Not a very auspicious start to my IBM career.

At the time "nail and string" pictures were a popular form of artwork, they consisted of a wooden "canvas", with nails knocked in, in rows, usually about half an inch apart, in some shape or pattern. Then the strings, or more likely coloured threads were joined in lines between the nails. The effect was somewhat spirograph like, since the lines often formed interesting curves. It occurred to me that I could write a program, using the manual that Dick had given me to draw patterns in a nail and string style on these old vector screens. This turned out to be quite pretty, but pretty useless to be honest.

Although, my objectives in this strange job were never really clear to me, but I made the best of it. For a start we had to find somewhere to live. We made full use of the IBM expenses, to start with in the posh “Chantry Mead” hotel. Then we downscaled to a rather dingy bed and breakfast place hosted by a landlady and her cats, although we did have our own room, we certainly didn’t want to stay there too long, she wasn’t the most hygienic of hosts!

Our problem was that we were about 21 years old, and didn’t really look even that old, so what sensible renting agency would hand over property keys to us two pesky kids It all seemed a bit bleak, especially considering neither of us had a driving licence at that time, and public transport in Hampshire restricted us to the expensive Winchester area. Well we were helped by my practical Mom and Dad, who simply brought down their tent and camping gear and pitched in the local recreation ground. Since it was August, the weather was fine and they treated it as a nice holiday, with a bit of house hunting thrown in.

They soon found us an Edwardian terrace near the Winchester Railway Station, and a rental agency, called Rosemary Gould Properties, were more than happy to hand us keys since Mom and Dad signed up as guarantors. The monthly rent at the time was £210. Which was affordable to us since we both had jobs, as Ann was working in Winchester City as a legal secretary. The only slight fly in the ointment, was that IBM hadn’t actually paid me any salary.

I had been told at interview that IBM was one of the few companies that paid their employees in advance, which was a nice perk. I’d been at work for more than four weeks, when I decided to pop over to personnel to query what had happened to my pay. Well when I got there, they were horrified that I hadn’t been paid anything. They immediately wrote me some kind of banker’s order cheque for six weeks salary, apologizing profusely. The reason I had been missed was related to my unusual start date in July.

This money helped a lot, since we had to pay our rent up front. One day I arrived at the rental agency to pay my rent, but they had just locked up. I banged on the door and a young chap surfaced on the other side of the glass. He shook his head pointing at his watch. I put my cheque against the window, and how odd, it wasn’t closing time after all! An early lesson that money really does talk.

Since I was a fresh graduate hire, IBM laid on several courses. The typing course was an hour a day for 30 days, to be honest it was pretty dull, particularly on the first few lessons, since we were not allowed to stray away from the home keys, so really not a lot you can do with just these keys ... a,s,d,f,g,h,j,k,l.

Maybe “lads had gas fads” !

I was one of the few graduates that stuck it out for the complete course, and whilst I didn't get close to Ann's 120 WPM, I did end up able to use all my fingers, and certainly was faster than at the start. I was relieved to be shot of the course though.

There were two programming courses, one on the REXX programming language, which had been developed by a recent graduate at Hursley, it was and remains one of the friendliest, easiest to use computer languages in existence. I loved it from the start of the course and made much use of it during my IBM career, particularly in the speech recognition days.

The other programming course related to IBM's mainframe line, and was called VM/370. The VM stood for virtual machine, this is a concept whereby each user of the mainframe computer (the 370 is the model of the mainframe) is given the impression that they are the only user. This was yet another thing that IBM pioneered years before anyone else.

The final course was given the mysterious title of "Physical Design". I had absolutely no idea what this was. When I arrived for instruction, I found that it was not a formal classroom based session, rather I was to be integrated into a team of technicians in a kind of pretend on-the-job training. It was an odd setup since I was being educated by a team of guys none of whom had a degree.

So, what on earth was Physical Design? Well, inside Hursley a large number of circuit boards and integrated circuits were being continually developed. A large proportion of the work is called routing, or wiring – this is the design of the copper interconnection of the many subcomponents on the boards and chips. Some of this work was automated by computer program, but quite often the program wasn't able to complete the full wiring, and the last 5% perhaps would require a human eye. The job was very similar to finding routes through mazes.

So, why was I, a graduate, doing what was predominantly a menial job? Well actually it wasn't all menial, whilst there I was shown IBM's special language to represent digital logic. I also picked up a simulation language used to verify chip designs prior to manufacture. Finally, I found out about our testing methodology to ensure that chips and boards came off the production line in perfect working order. All very useful experience for the years ahead.

One interesting aspect of this Physical Design course was that they all used the same old pinging IBM 3250 vector graphic display screens, supported by the department that I started working for. This way I got to see and use these screens for real for myself.

The guy that looked after me on this course was an enormous chap called Sam, he was probably 18 stone and well over six foot tall. He had a quite pronounced lisp – a heavily sibilant ‘s’, but was otherwise a gentle giant. He must have had a bit of a chip on his shoulder, having to train graduates, although he had had probably ten years experience inside IBM. Anyway his training method was to stand behind his students such as me, wielding a large cardboard tube used to hold technical drawings. He pretended to swing it at our heads like a baseball bat, but never actually struck us, at least not hard, but all the time explaining the intricacies of Physical Design.

At lunchtime Sam and his pal Conrad, or Con played chess, I got dragged into a few games and was about at their level, winning some and losing others, all with plenty of banter.

I can’t remember exactly how long this course ran for, it was a few weeks, possibly a month, but in a strange way I enjoyed it and got to like the guys there . This worked out well because in later years, when I needed their services to route my own circuit board designs, there was a mutual respect, which meant priority for me over other people in the lab.

So I finally understood why the course and department was called Physical Design, the answer logically enough, is that, my job was as a logical designer, and theirs as physical designers. Doh obvious really!

I remember nothing more of my time in the IBM 3250 vector display screen support department, apart from thinking that after six months I had achieved nothing and was starting to wonder if IBM was the right choice for me. Fortunately, out of the blue I got an offer of a new position working as a logic designer on a new range of display screens, currently in development. This was obviously right up my street, so along with three other graduates jumped at the opportunity, and switched department, but remained in the same office block.

This change of role must have been at the start of 1983. Good things come to those that wait.

Greenock and how to debug display planars

So, my first real product was a graphics terminal, well it was actually something between a visual display unit and a graphics card. A merge between the old world of display terminals and the new world of PCs. It consisted of three components: a 1980s IBM PC base unit, a square metal box about the size of a pizza box (we called it a pancake for some reason), and a display screen. The three components were piled up in a rather wobbly looking pile.

The idea was that it could be connected to a mainframe, as well as doing work locally on the PC. All the fancy graphics circuitry was on a large printed circuit board inside the pancake. The product had two names, an internal codename, “Aragon”, along with the “catchy” marketing name of PC3270/G. The PC needed a lot of connections to the pancake, I think it was about 60 wires, this required a very thick, black, ungainly, multiway cable with large connectors at both ends. This became universally known as the donkey dick!

OK, so our extended team in Hursley had designed the Aragon unit and had a handful of them working in our lab, but now we needed to scale up to allow mass production. The manufacturing site was to be the large factory in Greenock, near Glasgow on the Clyde. At this time most of the world’s high quality keyboards and screens were produced there, remember this is years before China took over. The plan was to produce a run of 100 Aragon units, to prove that they could be made reliably and quickly. What was needed in Greenock were engineers that could test the units and debug any that might fail. Fresh green Hursley engineers like me were ideal for the task. So my friend Leigh and myself were dispatched to bonny Scotland, our mission, should we choose to accept it, was to bring back as many of the 100 units – in working condition as possible, within a week.

I, by this time, had already built up a healthy scepticism about any kind of technology reliability, which for me translated into a flight phobia. Leigh didn’t want to go by train so we travelled separately. In those days IBM was super-generous with expenses, and actually insisted I travel first class, which was gloriously luxurious, the journey was nine hours and I was served meals by waiters with bow ties.

I passed the time doing a puzzle from Personal Computer World magazine, which involved finding a nine digit number with certain factors – it took up most of the journey, as I didn’t have a calculator with me – making it more fun. The scenery became progressively more spectacular the further north I travelled. Leigh and I met up at the hotel which was run by a miserable old Scot, who seemed to delight in being awkward, for example, when I wanted to phone home, he refused me an external line so leaving Ann wondering

what had happened to me. Leigh didn't like it either, so we checked out next morning.

We hired a car I think it was a white Ford Fiesta, I had passed my test much more recently than Leigh, but had more driving experience than him, so I drove and he navigated us from the hotel to the Greenock site. Every now and then he furiously pressed an imaginary brake pedal, this seemed a bit paranoid to me since I certainly wasn't speeding excessively.

When we arrived at security we had to get our badges authorized to get in, and we had to wait ages for anyone to host us. When someone finally did show up, they had no idea where to send us, and only with some long phone calls back and forth to Hursley did we finally get clearance to use a small cleaner's store room as a lab.

Then our pile of 100 large printed circuit boards arrived, and we realised we didn't have an oscilloscope in this small room which took us another hour to track down, again with impatient phone calls to Hursley and back. We had brought a test jig, so that each board could be mounted vertically, which at least made 'scope probing easier.

Well we tried the first board, hooked it up to a monitor, and nothing appeared. We tried the next one and this one at least produced some kind of picture, I can't remember the fault, but it didn't look easy. We tried a few more, it soon became apparent, that almost none of them worked. I think we did find one or two good ones on the first day. We settled down to debug some of the many bad ones, two or three of which seemed to share a common problem, namely that sometimes they powered up fine, other times, the picture was out of sync. We worked back trying to figure out the problem, and eventually realised that a counter chip wasn't being reset to zero at power on, so started at some random value. This was a bit of a disaster, since being a design flaw, it affected every board. For some reason the flaw had not surfaced on the earlier Hursley boards. So once again long fraught calls back to Hursley, to explain the issue, and get them working on a fix.

We called it a day and moved into our new Hotel, called the Redcliffe, which was a family run business overlooking the Clyde. The meals were delicious, and the rooms had a beautiful Victorian style with ornate taps and basins. I think they only had six rooms, so we were lucky to find two empty for our stay. They also had a proper switchboard so I could phone Ann directly at any time without bothering anyone. It really was luxurious, they happily fed us even when we turned up at 10pm.

Next day at work the Hursley team had already worked out, and sent us the details of a fix, for the counter flaw, which required a few patch wires adding, but fortunately no new components, which would have been difficult to

procure, and fit. Then we had to arrange to find a technician that had suitable equipment to add the extra wires to each of the 100 boards. There were plenty of people in Greenock that could do this for us but the bureaucracy meant that this took another day to arrange, and the full set of boards wouldn't be finished until after the weekend, although they promised that they could deliver about 15 rewired boards each day.

At this point we realised that there was no way we were going to finish in a week, so extended our trip to two weeks. Well these rewired boards were much better than the originals, but still a lot less than half of them worked. By the end of the first week we had only around 15 working boards, so even finishing in two weeks seemed very optimistic.

On one of the days when returning to our hire car in the Greenock site car park, I wasn't able to get the key to work to get into the driver side, Leigh had a try too and again no luck. We also tried the passenger door and again it refused to open. We scratched our heads baffled as to what had happened to the key, which looked absolutely fine. Eventually we noticed another white Focus a few slots further on, turns out we were trying to break into someone else's car! Duh.

A few of the boards were "write-offs" as during the motorised pass through the wave soldering machine, they had slipped off the side rails and dropped into the vat of molten solder underneath. When recovered from the bottom of the vat, using a wire hook, they looked really pretty since they were completely covered in solder both sides with every pin on every chip shorted together. To some extent these were helpful to us as they reduced our target of how many boards we had to fix! The Greenock team called them "submarines" which was appropriate to the Clyde !

Well on the second Monday we arrived bright and early, and fixed faults on about seven or eight boards, and we started to notice a pattern, in that quite a lot of the faults were due to a misplaced chip, either plugged in upside-down or one pin along from the intended spot. Now we reasoned that something must be going wrong with the pick and place machine that automatically inserts the chips into the circuit boards. So we brought this up with our Greenock host and found out to our horror that Greenock hadn't set up a pick and place machine for our boards at all and instead had recruited a few line operatives to hand place all the boards' components hence these misplaced chips. More phone calls to Hursley, which didn't really help other than to highlight the issue for the next batch of production.

Well it was getting late again and we still only had about 20 working boards and four days left. We considered extending for yet another week, but the Redcliffe hotel was full with a wedding, so we would be forced into going back to the miserable old Scot's horrible hotel. Hursley was also starting to get a bit

concerned because more delay was going to impact the schedule of several teams awaiting working Aragon display units.

At this point I had an absolute brainwave, I remembered seeing a layout drawing of the board, a few weeks earlier in Hursley, on my friend Geoff the wireman's desk. I gave Geoff a call and asked him to photocopy the drawing onto transparent overhead sheets, and mail them first class to us that day. He kindly did so and confirmed that they were in the post.

Of course next day post Hursley to Greenock was too optimistic, so again we plodded on debugging with the oscilloscope and got to about 25 working boards, by Tuesday evening. We had started working evenings but it wasn't helping much as we just got tired and tetchy with each other.

Well Wednesday morning arrived and we were greeted with an urgent delivery, inside we found Geoff's transparent layout drawing of the board and I excitedly aligned it over one of the failing boards and in less than a minute spotted a wrongly placed chip. Picked up another failing board and same again! Debugged two in two minutes!

By the end of Wednesday we had got to over 50 working boards without even turning on the oscilloscope. We were both overjoyed, since we knew the end was in sight and we could return home that weekend.

On the Thursday, our Greenock hosts offered us an interesting meter that beeped with a tone depending on the resistance, this was quite handy for tracking down short circuits caused by solder splashing around in the solder wave machine. It sounded a bit spooky, like a theremin, but did the business! We used this to fix about another ten boards.

By Friday afternoon we had got the total to over 70 working boards with about ten write-offs in the bin leaving less than 20 unfixed boards. I cheekily suggested to our hosts in Greenock, that we had deliberately left some failing boards ready for their in-circuit tester, which would eventually be used on the final production line to automatically find the board faults.

Hursley were very happy with this number of working boards, and told us that we were free to come home. Leigh and I went our separate ways, he by plane, and me by first class train again with a nice evening meal.

Eventually, long after midnight, I got back to Winchester Station and then into a taxi home to a sound sleep in my own bed, with my much missed wife, Ann.

Was the product a success, historically? Not really, like a lot of IBM products it was ahead of its time, it was an ugly forerunner to the graphics cards still popular today. I heard a later horror story that 20,000 monitors were produced without orders behind them and only a few thousand ever sold. Fortunately it

didn't matter since at the time the IBM PC was selling in the millions and along with profit on PC software, IBM was making vast sums.

So what is the moral of the story? Certainly that there is more than one way to skin a cat, and doing what you were taught at university isn't always the quickest way. Oddly for a few days after the trip I had a slightly Scottish lilt to my Brummie tones.

Tales from Millbrook

My first job in Storage, or Storage Products, to give it the correct title at least back then, involved relocation to the IBM satellite location at Millbrook. Millbrook is on the outskirts of Southampton, also on the coast, but really a grimy suburb with a gas works and a sewage works as the main highlights. There are many out of town retail outlets, plus various industrial sites. The IBM site was originally intended as a retail outlet, and had a metal construction, with massively high ceilings. Birds frequently got stuck up there.

The space was broken down into a few corridors of offices, probably about 60 offices in total, and a large test area, plus a smallish cafeteria.

The cafeteria could boil water, but not much else, so they were supplied by the much larger Hursley catering section about 25 minutes away, so the meals were always a bit second hand.

One day we queued up for lunch, I was about 4th and had a main course, plus apple pie much like my three pals up ahead. I was about to pour on my custard, from one of those metal jugs reminiscent of school days, when I hesitated looked more closely, and exclaimed "this custard looks a bit green". I took a sniff and it really didn't smell like custard – the two dinner ladies were both characters and one of them rushed over to help me, whilst the one on the till asked what I was moaning about now. They thought the consistency was wrong, so one tried a teaspoon full – aagh – that is the pea soup, which had gone missing earlier that morning. So they brought out the proper jug of custard, and I saved ruining my apple pie. Tony and Roger were given a fresh slice, but Gordon said it didn't matter, he didn't mind and ate his anyway. He is from Scotland though. Go figure.

Millbrook was a nice community, with about 70 IBMers working there, so we were much more like a small company, although we still had a couple of security guards manning the gates at the entrance. The car park was a line along side the main building visible to the guards. One day I had arranged to have my Marina serviced and MoT'd. The chap, Mark, who did the work was happy to pick up the car from a pre-arranged spot, and I just had to leave the keys under the seat and a door unlocked usually just for an hour or so. The time came and Mark walked along until he found my car, grabbed the keys, and drove it off. Only trouble was the guards were on to him and they swung the gate shut trapping him inside. Meanwhile I got a phone call to my office saying "Bob, someone is pinching your car". I explained it to them and of course they let him out and on with his business.

Reflecting on this we realised how much better security is in a small company, where everyone is known personally.

So what was the day job at Millbrook like? Well my role was called a microcoder. This is a historical term, and relates back to days when disks were controlled with sequencers long before I started. The work was now a more traditional programming job, since disk controllers had moved to microprocessor control. At the time of my recruitment to Millbrook, we were between projects, one, internal codename, Kestrel was already shipped and selling. The other, internal codename, Harrier, was just starting development.

Since I had just started, it made no sense for me to work on the older Kestrel project, rather my focus was on the new Harrier Controller. The project was developed from the ground up, both hardware and software, including the design of the disk drives, internal codename Redwing, of which up to four could be installed in each Harrier Controller drawer.

The Millbrook team had responsibility for the Harrier and the Hursley team looked after the Redwing drives.

So I was in a team of about eight programmers, each responsible for parts of the Harrier Controller software.

My parts were all related to error recovery, which is a general term for automatically sorting it out when things go wrong.

The three detailed sections I was responsible for were:

1. Block error correcting codes.

2. Link error recovery.

3. Block Reassignment

I'll explain each in turn...

The error correcting codes are used to correct minor errors when reading a block of data from the disk. The error correcting codes work by appending several extra characters to the 512 characters in each user block. These validate the 512 characters, and if a few characters read back wrong, then they can be corrected using a trial and error algorithm. If more than a few go wrong, then it can't be corrected, and data loss occurs. I was lucky here since the design of these codes had already been established before I started, based upon research work from an IBM mathematician in the US. I just had to follow a recipe to write the program using the research paper and I also had a Pascal program which one of the Millbrook engineers had written. All I had to do was translate it, line by line from Pascal language to assembly language. It didn't take much longer than a few weeks. The thing I remember about this program, was that once I had fixed a few bugs and typos, it sprang into life correcting errors properly and it dawned on me that it was then bug-free since the coding scheme was mathematical it either works perfectly or not at all!

The link error recovery scheme was designed to correct errors during transmission along our serial link. This wasn't based upon codes, rather it was a checksum and retry scheme. Most of the time the checksum would tally confirming successful transmission, just occasionally though an error would occur, the checksum then didn't tally, and so that data was discarded and retransmitted. A simple analogy is asking someone to repeat themselves, when you don't hear the first time. It was all pretty new at the time and my name was put on a patent which we used to protect our idea. I got a few hundred quid as an award for this work.

The block reassignment scheme was intended to deal with scratches, or tiny bits of dust getting onto the disk surface. We had about 1% of the drives blocks reserved as spares. When a block got damaged by dust or scratches, my code had to move the blocks around to make use of a spare block instead. My office mate Tom (not to be confused with our son, also Tom) helped with this program. It took two or three months to plan it all out, and about the same again to write all the lines of code. During the planning phase we needed to really concentrate, but other people kept on interrupting, so eventually Tom put a sign on our office door, marked "Do not disturb – R. I. P. (Reassign in progress)!". We were then left alone and able to complete the design.

So a little over a year had passed, and the Harrier software was more or less complete, but untested. Why untested Reason for that was that the hardware was not yet ready, although that was now imminent. We did have some simulation software tools, which we could and did use to try out our software, but this was slow and not a very realistic model of the intended hardware, so often a lot of time was wasted struggling with the simulation rather than finding the software bugs we needed to flush out. I preferred round table

reviews, where the other programmers in the team met and communally read through the software looking for bugs by brainpower. It sounds really tedious but they were light hearted and people often used to joke around, and they were limited to an hour or two.

Anyway the Harrier hardware started to appear, and gradually we started testing our software by installing it by programming EPROMs.

EPROM is an acronym for Erasable Programmable Read-Only Memory). It is a programmable read-only memory (programmable ROM) that can be erased and re-used.

Once the controller software was capable of reading and writing the Redwing disk drives, we settled down to a long phase of formal testing. The way this worked was a test team ran various tests and when they discovered a bug, they would stop and print out a system dump which was about thirty or forty A4 pages of numbers. Then the tester wrote up the details of the problem into a mainframe computer we all shared, the problem was allocated a number, and this number transcribed onto the printout of the system dump. The printout then dropped into the mail slot of the programmer most skilled at fixing that particular problem.

There were hundreds and hundreds of these printouts floating around, and they often went “around the houses” endlessly passed between programmers each in denial about the bug being theirs!

This went on for, I think, two years, it sounds like torture. I really loved it since I was being paid to solve puzzles all day long. It was a bit competitive, and I enjoyed proving my programming peers wrong, and trying not to get caught out with bugs in my own programs.

Tom, my office mate, came up with a way to squash the software down to fit the entire listing in one printed book about 700 pages long, it had to be reprinted every few weeks, otherwise it wouldn’t include all the latest bug fixes. So it was a bit expensive in paper, but this helped a lot with the debugging process, as screens in those days couldn’t display many lines of text.

So we used simulation, review and formal testing. Eventually we were done and no serious bugs remained. We shipped Harrier and total product revenue was something like four billion dollars, although not all of that was directly attributed to our teams for some accounting reason.

I heard that the support phone was covered in dust. That was a nice tribute to our work.

Too high, or too low?

http://www.everytrail.com/view_trip.php?trip_id=662882

The web link above has a series of photos that recreate a hike I did, certainly pre-Tom, probably late-1980s. I'd not long joined Storage, and they were at that time, keen on team-building, so I was sent on a "management" course extending over four separate weeks, three of which were psychological BS in local classrooms that were eminently forgettable and forgotten.

The highlight, however, was an outward-bound course in the Brecon Beacons, this struck me as an opportunity for a five-day holiday, albeit away from Ann. There was some abseiling and rock climbing days, which were fun and scary, and also a "challenge" where, after a long hot walk we had to cross a stream to reach the small coach to take us back to the caravan. We were given no maps, so had no idea if a bridge was nearby, I wasn't happy as I knew this was going to ruin my walking boots, in the end we dashed across and, no surprise, arrived with soaked feet. It did ruin my boots, but I got my own back by leaving them crashing around in the site tumble dryer!

The best day, however, was the hike to find the 1943 WW2 Wellington that had crashed not long after taking off from Wellesbourne airfield (Eileen and Roy had recently moved there I think).

The setup was that on the previous evening we were given maps, bearings, pressure readings, etc. but crucially the pressure reading given at Wellesbourne was wrong by some amount, and the idea was that we could work out the flight path height by taking into account the pressure error, and intersect with the contour lines on the map to locate the crash site. When I did the calculation though, I noticed that the error was the wrong way, i.e. they would have been flying 300ft HIGHER not lower, I checked and double checked in that freezing (it was October) caravan for most of the night. In the end I decided to go along with it and moan to the outward-bound expert, himself an experienced glider pilot.

Well, next morning we had an entirely different problem: thick fog, just driving us to the car park was a crawl all the way, and the hills were even worse. I was navigator, and I was a complete nightmare, refusing to let anyone move until I'd proved where we were. The only landmarks are the few patches of water, not much bigger than ponds and the slope angles which can, in theory, be matched to the contour lines on the OS map.

When we got to the nearest bit of water, I figured out a compass bearing to the crash site, and a distance, which I insisted was counted by paces, and the angle maintained using three sticks pushed into the ground to guide us as we fumbled about in the fog, eventually I noticed I was walking on bits of rusty metal. Then we found the memorial, and we knew we had successfully completed the task. Yay! We paused for thought, reverently thinking about those poor airmen all those years ago.

On the walk back I asked the outward bound course guy, a bearded Welshman, about the error on the paperwork, I distinctly remember his reply : “a lot of people say that”.

There was another IBM team that did the same exercise the next day, in clear sunshine, they turned up late for the evening meal frustrated not to have found the Wellington.

Well that was that, we left the Brecon Beacons, and I returned to work the next Monday.

But, it niggled at me so I wrote up the entire exercise and appended it to the nascent Hursley Gliding forum, an IBM forerunner to internet newsgroups.

Hursley has a very high density of pedants, so it wasn’t long before I had at least half a dozen responses, and not all were on my side, but eventually there was agreement that the exercise paperwork was wrong. I suppose I should have printed it all out and posted it to the Welsh course guy, but I didn’t get round to it.

So, I thoroughly enjoyed the experience, and took a lot from it, although ironically management (the intended aim) was never of interest to me. The problem solving aspect is far more fun. Debugging drives has a lot in common with air crash investigation, but is surely easier and certainly less of an emotional drain on the soul.

Havant a clue

It was 1992. The world was changing in lots of ways: IBM was slow to respond to the new digital world, the PC market was being eroded by clone makers, and unsold inventory. The mainframe computer world was thought to be doomed, profits were down and IBM’s job-for-life concept was rapidly consigned to the bin, as various packages and even forced redundancies started to bite. Satellite locations like Millbrook and Eastleigh were sold off,

and large sites like Hursley told they had to focus on one thing in that case it was software.

What did all this mean for me? Well I was doing pretty well in the storage team, which had had some notable recent successes, very high revenues and profits, so that was all good, problem was that storage is very obviously hardware, so no way would we be moving from Millbrook to Hursley.

Question was where were they going to put us? The only place left for the storage team, about 100 strong was Havant. Very few people wanted to go as most of that team were living within a few miles of Hursley. Winchester to Havant is about a 30-mile drive, 33 on an expense claim! The drive is a miserable motorway, busy and frequently blocked with accidents, there is a back way but that is slow, usually an hour at work times.

So to encourage people to transfer there was an incentive scheme based on salary bonuses: 5% the first year, 4% the second year and so on. Since this was a contract change each employee had to sign up. Some people were lucky, for example my boss lived in Waterlooville, which was a few minutes from Havant, so he signed up for the free money right away. Some people decided to move house, so they signed up, others were perhaps on the south side of Hursley, so decided that the commute was tolerable. Some contractors left, one or two people near retirement decided to go a little earlier than planned. Eventually, all but a handful had signed, so the pressure was on the managers to get those last few signatures. Well my boss knew it was hopeless trying to force me, and his boss tried but knew me pretty well too, so I had a visit from the executive boss above him, and despite 15 minutes of pressure I simply wouldn't budge, and he left my office, with tail between his legs.

My position was that I was prepared, and even happy, to work at Havant for no more than a year, whilst I was looking for a "career move" preferably to Hursley. I had an ace up my sleeve in this regard as I'd always stated that I was open to a move to another role in Hursley on my annual appraisal form, which I gleefully pointed out had been ignored for the last five years. I had included this as a kind of insurance policy for just such an eventuality.

So I never did sign up for the bribe and we(me included) all moved from Millbrook to Havant. Whilst I enjoyed working for storage, doing that commute for years was not acceptable to me, so I pressed on, but with an eye out for jobs in Hursley. Trouble was they were sacking people in Hursley, not recruiting them, so really I had to bide my time.

One example of a reason I hated the motorway commute, was as follows. It was a boring, drab, rainy morning I was sat in my blue Rover 400 in the middle lane at a pretty consistent seventy mph, gradually passing hundreds of lorries grinding along nose to tail on the inside lane, when I heard a horrible sounding

clonk from under the bonnet. Obviously at that speed there isn't a lot you can do, but terrified, I braced myself and started trying to find some way through the massive freight snake to the hard shoulder. Immediately after the clonk, I felt something solid hitting the underside of the chassis, i.e. below my feet. The miraculous thing was that the engine continued running as if nothing was wrong. So tentatively I continued on wondering what on earth had happened, I noticed that the red ignition light seemed to be on. This didn't look right, and then I remembered that comes on when the battery isn't charging. Maybe the fan belt had broken and flew off, but that wouldn't really explain the clonk noise. I decided to press on to work hoping that the battery would last at least long enough to get me there without the lights dimming or anything else that needed 12 volts failing. Well I arrived at work, lifted the bonnet and sure enough the fan belt had gone, but so too had the pulley that keeps the belt in tension. The bracket had gone too, and it looked like the retaining bolt had the head sheared off leaving a broken piece of bolt inside an inaccessible hole. I dropped into the office and asked a few locals for a garage recommendation , then immediately dropped the Rover around there explaining the damage. It was something like December 22nd, so I had little hope of getting it repaired for Christmas, which was going to be a right pain. However the garage man fixed it by the next day for a very reasonable price so I only needed to cadge a couple of lifts from a friend. Even on one of these lifts we had a near miss, narrowly avoiding shunting a car on the same motorway. The journey was hateful.

The storage product we were working on was one of the first disk drives with the tiny, at the time, 3.5 inch diameter disk platters inside. The internal codename for the unit was "Allicat". IBM in Rochester, Minnesota had developed the product with a parallel connection to the outside world. This connection usually took the form of a rainbow coloured ribbon cable with 40 wires about two inches wide, and no more than a couple of feet long. The cable was a little unreliable, and using more than about say 10 such cables wasn't really practical. In Millbrook we had long known this, and had pioneered a high speed serial cable instead. This cable was much neater, only about 1/4 inch diameter, and could run to several meters long if needed. Also hundreds of drives could be reliably connected to a computer, typically in a rack about the size of a large fridge-freezer unit.

So our job was to make a serial version of Rochester's Allicat disk drive, which became known, sensibly enough, as "Serial Allicat". To do this required a new quite complex chip to be designed, and this needed two or three small teams of logic designers, analogue engineers plus people familiar with simulation tools, because you don't want to have design errors in a chip, since correcting a chip could take several months. So the chip is simulated on a mainframe computer to check it works before production. I helped out in the chip simulation team for a few months.

This chip had the rather fetching name, I kid you not, of the “SIC chip”. The SIC stood for “Serial Interface Chip, and even that horrible name doesn’t really work as the astute reader will note that the “C” for chip is used twice in “SIC chip”, but once these things catch on there is no going back, and it isn’t a name that would be disclosed to a customer, so remained a “sick” joke.

As well as this new SIC chip we needed a few changes to the Rochester software running on the main microprocessor in the Allicat drive. So a few programmers were also needed, I was in this team and my job was the module that retried transmission when the serial connection had some kind of error, such as a glitch caused by lightning. The software was called the link error recovery routine, it wasn’t a huge job by any means, but was interesting because this was the first time we had used a high level language in disk software. The chosen language was the, still popular, “C” language invented around 1970. All previous disk drives had used assembler language, which boils down to machine code.

The first few chips were due back from IBM Manufacturing which I think was Burlington, Vermont, pretty much on the day of our move from Millbrook to Havant, which was a bit inconvenient. I remember saying that I wanted to be debugging on the lorry moving us, and definitely wanted our kit to be last on the lorry first off.

First chip power up is always a nerve racking moment, because it always seems to be plagued with false alarm problems, that make people fear that the chip doesn’t work. The test is often called the smoke test, because after applying power that is the first thing to check for. Typical false alarms problems are things like power supplies wired to the wrong pins on the chip, or reset circuits not working correctly. Managers are usually running around all with the same question, namely, “Does the SIC chip work?” Our test was complicated by the Havant move, but although it might have had one or two minor false alarms, was pretty much working properly, and it was only a few days before we were transferring data on and off our first serial Allicat disk drives. Even at this early stage we were pretty confident that we would be able to mass produce drives without requiring another pass at the chip design.

Our work for the next few months would be to test out a few hundred drives running continuous writing, reading and verifying of data. Loads of minor bugs usually surface during this phase of the project, and the vast majority can be fixed by changes to the drive software. Problems divide up into those that affect all drives(generic) and those that affect only one particular drive(specific). The generic type is obviously of higher priority. If a problem only affects one drive, then one easy answer is to scrap it before it ever gets to a customer.

After a few months we got to the point where all but one or two generic problems remained, and perhaps 10 drives didn't work at power up, out of say 500 drives in total, so we were in the process of ramping up for mass production.

OK so now I need to explain a little about the drive test area in Havant. There were about six test cells, each about 12 foot square, all in a row with Perspex panelling between so you could see from one cell to the others. A corridor ran alongside. The entire area was on a raised floor so that services like power, air conditioning, plumbing, etc. could run underneath. The floor consisted of large, heavy, wooden tiles each about two feet square. Some of these tiles were perforated with a large number of half inch holes, to help with air circulation, heating, etc. The area was quite noisy, both from the drives on test, as well as the air conditioning. It was also cold, particularly in winter, as it was at the time. What annoyed me was that when I was working in one of the cells, if I was sat above one of the perforated floor tiles, the cold air would blow up my trouser legs and I'd end up frozen. I considered cycle clips but in the end got the tiles swapped between cells, so that the perforated ones were in the cells I didn't work in very often.

So one of the last remaining generic problems was that a block of data on a random place on a random disk drive would be occasionally be unreadable. Nobody had a clue why, but everyone agreed this problem was a killer.

I did a careful analysis of these unreadable data problems, and found that firstly they were extremely rare, certainly less than one bad block in 100 million blocks read back. There was no pattern to which drive failed, there was no pattern to which block failed. The only slight foothold we had was that the unreadable data was always made up of two sections, firstly a portion of the correct, expected data and then a section of what looked like the previous data. It looked as though the writing operation had been interrupted part way through. Typically a rack containing about 100 drives would work properly overnight, apart from one or two drives that failed with this unreadable data by the next morning.

We needed to recreate the problem on an instrumented drive, we had two or three drives that had logic analysers attached that could trace the flow of the drive software. Unfortunately even after several weeks of trying, I couldn't get the problem to happen on these instrumented drives. So we knew the problem existed but were completely powerless to fix it.

The mass production of serial Allicat drives was building up, and we were really in a bind since as far as I was aware every single drive was susceptible to this unreadable data problem. An issue up with which our customers would not put !

Havant manufacturing when faced with this sort of problem put in place what they call a manufacturing screen. The idea is to test every drive for the failure and screen out the drives that fail the test. This sounds good, but the reality was that the test would take months to be conclusive, and anyway my opinion was that eventually all drives would fail. Anyway they pressed on and had two piles of drives, one marked “passed” and the others marked “failed”. All nonsense as it later transpired.

Meanwhile I continued try to recreate the problem, I racked my brains trying to figure out the difference between the instrumented drives in my test cell and the racked drives in the other test cell. The only thing I could think of was temperature, my test cell was quite a bit warmer, since I had removed the perforated floor tiles. Aargh – hoist by my own petard. So on one overnight run I got hold of a few office fans and left them blowing cold air at the instrumented drives. Sure enough next morning – hallelujah – the analyser had triggered. It didn’t take long to work out that the secondary drive processor that controls the position of the disk heads was getting reset randomly. It was supposed to be reset by the main processor in case the secondary processor crashed or got stuck in a loop. Problem was quite a blunder in that the reset pin was set as input when it should have been an output. Result being that the reset pin was floating, and so could happen at any random time and if the drive was writing data at the time, then the heads would retract leaving the partially written data block which was then unreadable data. The symptoms fitted perfectly, floating inputs often vary according to temperature.

The fix was an easy one line change to the drive software, to make the reset an output, which once applied meant all drives ran same test for months without error. My third level boss Stuart came down to my test cell patted me on the back, shook my hand and told me that I had saved the company about four million dollars in Allicat drive inventory, which was now saved from landfill. We could now confidently ramp up production volumes and start shipping to the field. I’m not sure how many Serial Allicat drives we sold, but it was probably in the hundreds of thousands. Certainly a successful money making project.

The moral of this tale? Certainly not giving up was key here. What I took away though was the interesting observation that problem recreation is actually most of the battle, debugging was relatively easy thereafter. Put simply you can’t solve a problem that you can’t make happen.

So I still enjoyed working in storage, but the two hours wasted every day on the Havant commute was too much – Tom was a toddler at the time – so I continued asking about new roles in Hursley. I started realising that I’d have to take anything, and put up with it, hopefully to be able to move position to something better again once back at Hursley. That was pretty much how it worked out, but that is another story.

Retail Therapy

I continued the miserable Winchester-Havant commute, and was in the slightly awkward position of having to avoid working on exciting new storage products, since it was widely known my time in storage was limited. Finally, my boss told me about a Hursley job offer in the area of Retail Systems, this department did software for point-of-sale systems, more commonly known as cash tills. I knew very little of the team since they used to be located at the Eastleigh satellite lab, but they had very recently been reabsorbed into Hursley as part of the sell offs to save money.

The job description didn't sound very inspiring, but it was programming, and it was the lifeline back to Hursley I was looking for. So I feigned enthusiasm and agreed to an interview with the hiring manager, who was just a name to me. I showed up and the interview was straightforward, the manager was himself considering taking an early retirement package and he just needed a programmer to fill the role, and he seemed happy enough with me, so somewhat unceremoniously, I got the job, and was told to show up on the 6th of the month as is the IBM tradition.

When I got there it struck me as quite exciting, since I knew nobody, so it was much like starting a new job. I was introduced to my office mate Chris who was a few years more senior, and assigned to be my mentor as I found my feet. Chris had worked in customer education before and during his time in Retail Systems, so was an excellent communicator, nice guy and seemed ideal to show me the ropes.

It was great to be in a new block in Hursley, I'd arrived half an hour early, since I was still used to getting up for the Havant commute. Well Chris spent most of the morning explaining the setup and it didn't seem too bad. The main thing that he told me was that the department operated on the basis of short term contracts, some were with retail stores, like John Lewis, others were with other IBM locations, such as IBM Raleigh in North Carolina. Either way all my hours had to be recorded against a job, so that our funding was paid for. I was still a full time IBMer, and paid my annual salary, but from the department's point of view I was a lot like a contractor.

The other big change was the infrastructure, I was used to working in teams such as hardware, testers, architects and planners. Here, I had to be my own boss, arrange and agree my own contracts, design the software architecture, write it and test it all on my todd. This seemed a bit impossible, but Chris assured me that the typical retail project was much smaller than designing a disk drive that I was used to.

My first job was to be working with IBM Raleigh on a set of device drivers for a new IBM touch screen based cash register, that had been ordered in large volumes by the fast food outlet called Burger King. The idea of the touch screen is to offer protection against spillages as well as having a programmable keyboard that could be changed to accommodate menu updates and offers.

Mention of spillages reminds me of an incident in the Hursley cafeteria. The cafeteria is laid out in a vast grid of tables, seating well over 1000 diners. At the end of the row that I regularly sat at was a cold water station. This station had a looped tap at the back, and a small lever operated fountain at the front for just grabbing a quick sip of water direct from the spout. An American visitor arrived at the water station, plonked his tray on the fountain, and put his glass under the tap at the back. Unbeknownst to him his tray pressed the button for the fountain, and whilst he was trying to figure out how to make the rear tap work, water was squirting from the fountain onto the underside of his tray and from there pouring onto his smart, pale, suit trousers. He looked across at us, confused, looking for help with the back tap, but seated as we ate, all we could do was giggle and point, flapping at his trousers, which confused him all the more. After a while he must have felt the water soaking through and stepped back, cursing. I really hope he didn't have to do a presentation that afternoon, because it looked for all the world as though he had wet himself. I never saw that happen again despite eating there for years before and after. Anyway back to the description of the till.

There was a two row alphabetic display screen which presented items and prices to restaurant customers, along with a magnetic swipe card reader to take payments using credit cards. Each of these also needed device drivers writing.

What is a device driver? Well it is a layer of software that sits between the customer (Burger King) application, and the electronic components that make up the parts of the till.

Chris and I sized up roughly how long each of these device driver programs was going to take to design, write and test. The total was getting on for about four months. So I had a nice self-contained project to get on with.

I quite enjoyed this little contract, it certainly was very different having to write the specifications as well as do the programming, but there were some nice benefits. I could, for example, change the specification to simplify my task, without having to go cap in hand to an architecture team. In other words I was in control of my own destiny!

When the job was pretty much complete we noticed I was running out of time to complete the final testing, so Chris offered to help provided I gave him a list of tests to run. He insisted that I write up an itemised test plan, and he showed me how to get little square boxes printed next to the description of each test. He had a mantra that he recited with a laugh, namely, “Tick in box”. The idea being that we were not finished until the test plan had all the boxes ticked. Very simple idea, but very effective. Between the two us us we managed to finish the project more or less on schedule. Raleigh were happy with the work, and took it over to fit in with some larger software scheme they had in mind.

I later heard that some Mexican programmers had been hired by Raleigh to do a similar job, I think on a different till. They were on much the same schedule as Chris and I, but they had failed to deliver anything usable so Raleigh had to rush together the job themselves. All this made us shine all the more.

Somehow, Chris managed to wangle us a posh night out to celebrate. We took our wives, Gale and Ann to Lainston house and had a slap up meal.

Well this was the highlight of my time in Retail Systems, and I can remember nothing of other jobs I had whilst there, and I think there wasn’t as much work coming in to the department as there were people. So that role was looking a bit bleak, and I started to look for something else again.

“Wreck a nice beach” or “recognise speech”

Hursley has a long tradition of doing a few maverick style projects, things that were not going to make a fortune, but were interesting technologically, in fact for many years there was a department called “Special Engineering”, or SPENG for short. One project that came out of this department was a screen for visually impaired programmers, called “Parrot”, this unit had one of the first synthetic speech systems in the world, and was capable of reading aloud the screen content for the user. We had a couple of completely blind programmers in Hursley, and they were able to work using these units. Sadly, the SPENG department got closed down during all the cut backs, but somehow despite the financial constraints, one or two engineers continued working in this nascent field of speech technology.

Now everyone has to be somewhere in IBM, so these speech experts had to be in a department, and the only thing left was the Retail Systems so that is how they were funded. One of these engineers was a gentleman by the name of Dr Eric. He was a physical chemistry Ph. D. He had worked at Harwell studying the decay of neutrons amongst other things. In IBM he had worked on the chemistry of the surface coating of storage drive platters. More

recently he had been on assignment to the US to look into research on speech recognition. He was now back in Hursley, and working on the British dialect of English, something of little interest to IBM US.

He was overloaded with speech recognition work, and was looking for an accomplice . This struck me as a perfect opportunity, and I was champing at the bit to meet him for an interview.

I even had some speech recognition experience, albeit a decade earlier, as my final year project was “A spoken digit recognition system by microprocessor”. This was at least something I could talk about at interview.

A kindly chap called Peter who was quite senior in IBM, had some responsibility for keeping an eye on the speech experts, he was in the Retail Systems team and knew me from the Burgerking contract. So I discussed it with him and he arranged for me to meet Eric, although, I do remember him warning me that Eric could be a bit of a character – he struggled to try and explain what he meant, and in the end said Eric could be a bit loud or wild even, but was at pains to make clear that he was perfectly amiable once you got to know him.

Well the time came, I think the interview was around 10:30, I found my way to Eric’s office, as was greeted by a thick mop of ginger hair and a spectacular moustache. We shook hands and I sensed immediately that interviews were not really comfortable things for Eric. So I kind of instigated the process by introducing myself, and explaining that Peter had discussed the role with me briefly, and that I was here to find out more. Eric told me straight up that he wasn’t a manager so wasn’t able to do hiring/firing stuff. I acknowledged this, but of course Eric’s recommendation was going to be key, since Retail Systems management had no experience in speech.

Then he explained about the speech group being hangers on to Retail, and that most of the work was being done in the US, although they were starting to get set up in each of the large countries in Europe, since creating a German speech recognition system, for example, was not just a matter of translation.

I told him a bit about my final year project, he was interested and asked me a few details, but whilst it was something, it was clearly not very impressive compared with IBM research ten years on.

Then I started asking Eric how the IBM recogniser worked, and his explanations were not satisfying me, so I started asking him to show me examples such as the phonetic alphabet that was used, and how the system converted the speech signal into phonetics, the more he showed me on his screen, the more it left unanswered questions. Quite a lot of my questions started, “wouldn’t it be better if ...”

The interview process had become completely reversed, I just pressed and pressed for more and more details, and Eric kept on answering as best as he could. It would probably take a full day to give just an overview of the IBM recogniser, but I wasn't aware of that at the time. Eric just kept on trying to placate me and gradually the complexity started to dawn on me. This made me even keener to get this job, it seemed fascinating and perhaps somewhat arrogantly, I assumed I could make the system more accurate, which is the holy grail of speech recognition.

I felt some pangs of hunger, it seemed that Eric brought sandwiches, so wasn't bothered about finishing or at least breaking off the interview for lunch at the cafeteria. I surreptitiously glanced at my watch, shock, horror it was after 2pm and the cafe was closed so I'd missed my lunch. I realised that my job prospects would be better if I wound down this everlasting interview. So I asked when I would find out if I had got the job, and again Eric explained that he had no idea what went on in the heads of Retail Systems management and basically had as much idea as me. Well that was about it, I thanked him and went off in search of a bag of crisps and chocolate bar in lieu of lunch.

I waited patiently for a few days, with nothing much happening. I checked with my immediate boss who knew nothing. In desperation I tried Peter, and he said he would try and find out. I don't think I ever formally got the job, since I already worked for Retail Systems, there wasn't a transfer of department or even manager needed. Anyway Peter told me that he had had a word in a few ears, and I could move into the spare seat in Eric's office. Apparently no one from other Retail areas was that keen to share with Eric because of the noise of speech sample playback, and recordings.

I could move in the following Monday morning, which I think Peter just made up. Looking back I suspect that, after the interview, Eric had simply carried on working and no-one else did anything. Anyway none of that mattered, Monday came and I brought my cupboard's worth of stuff and simply moved in, and carried on with my interrogation to find out how the IBM Recogniser worked.

I can't remember exactly what I started on, my main role was to create some specialised language models, that were aimed at application areas suitable for dictation. I think I rebuilt the UK radiology model, which was the first proper speech recognition product. It worked pretty well since the vocabulary is restricted, and radiologists are used to dictating. After that I was involved in three other models, namely, pathology, legal and medical. These varied in success. Pathology for example was stunningly accurate, but even if we sold a copy to every pathologist in the UK, it barely paid for my time to create it. Legal was a bit disappointing, the language used isn't that different from general day to day English, which was too varied for the accuracy of our system at the time. Medical was better but still the sales volumes were not great.

We needed to improve the accuracy of both the acoustic system, and the language model, and most importantly, offer general English suitable for anyone wanting to dictate a letter or write a report.

So began what was to be seven years of fascinating research and development into the field of speech recognition.

I'm not sure what the moral of this tale is – perhaps it is best to assume you got the job until you hear otherwise.

Recognised in dispatches

After our moderate successes with the specialised language models for speech recognition including medical variants and legal, we knew that a general English model was the "next big thing". This would appeal to a much wider audience, and hopefully make speech recognition a profitable business.

We needed to improve three things, I'll describe each in turn...

Firstly we needed to improve the microphone and soundcard used for voice input. We had previously charged doctors and lawyers a few hundred pounds for a custom designed IBM sound card, this outlay wasn't going to appeal to a wider audience. We were quite lucky in this area because at just this time gaming machines were standardising on Soundblaster sound cards, and some were even integrated onto the motherboards of the latest PCs. These were high quality and much cheaper than the IBM cards. All we decided to do was to supply a reasonably low cost head set microphone in with our product package.

This meant we could predict and control the microphone volume level, and keep it near to the user's mouth.

Secondly we needed to improve the acoustic model, but what is the acoustic model? It is really a layer that converts the sound wave from the sound card into phonetic units, typically one hundred times a second. This model is built automatically using known recorded, spoken sentences from a variety of accents around the UK. We decided to increase our database to about 800 people each reading about 100 sentences. This would also be required with separate recording for each of the European languages. We would need to recruit a huge number of volunteers.

Finally we needed to improve the language model. This had three separate sub-jobs :

1. To collect a few hundred million words of typical UK text, best source for this is from broadsheet newspapers. This is known as a corpus.

2. To decide on the vocabulary, we went, initially for about 30,000 words.

3. To include all alternative phonetic spellings for each of the 30,000 words. This is needed because some words have more than one pronunciation, for example, the word “either” can be spoken with a starting “eye” or a starting “E” sound.

The 800 recordings for the acoustic model seemed interminable, even if we could find say five volunteers per day, it was going to take over six months. We did offer a raffle prize to one out of each batch of 100 volunteers, but it didn’t help much. The main way to recruit was via cold calling around the Hursley Labs. Ideally we wanted 400 gents and 400 ladies, but the demographics of gender in Hursley made this a dream and we settled for something more like 500 gents and 300 ladies.

Eric and I took turns phoning people up to try and persuade them to spend about forty minutes reading our predefined sentences into a posh, desk-mounted Sennheiser microphone. We used this posh microphone on the understanding that the better quality sound we had, the better our acoustic model would be. Also we could easily lose quality, but it is impossible to gain it.

Each volunteer would be invited into a quiet office, with just the recording equipment and a PC present. There they would be shown how to use the mouse to move forward through each of the sentences. They were instructed to be careful not to speak until the microphone was turned on, and not to turn it off until a little after the end of each sentence. Finally, they were shown a back button for use in the event of fluffing a sentence. We made it as easy as we could.

After giving these instructions hundreds of times, I started to forget who I’d said what to, so I think I found myself repeating things just to be sure. Probably appearing a bit loopy!

The vast majority of the volunteers had no trouble recording their sentences, but occasionally people had trouble. One blind volunteer asked that we email him his sentences, so that he could print them out on a braille printer. This was a great idea, but for the fact that at some point he must have slipped out by a sentence, so that part way through, his sentences corresponded to the previous one. Luckily I checked on him halfway through and was able to rescue the situation, and after that I stayed with him to help, which of course, with hindsight, I should have done throughout.

We also had a very kind lady, unfortunately profoundly deaf since birth, it must have been a real effort for her to try and read the sentences. Her speech was very difficult to follow. She did complete them, but the reality was that we couldn't really use them, since most sentences would not align her sound recordings with the expected phonemes. She was also hoping to be able to use the speech recognition system, when we had it complete, but realistically it was never going to work very well for her. All a bit of a shame and rather sad really. Looking back, I really wish she had won the raffle, but nope life isn't like that.

We asked each volunteer, what their normal accent was and, along with their names added this to the database . The UK is a country with one of the most diverse ranges of accents, it isn't hard to find twenty quite distinct accents or dialects. To make recognition work well all over the country we really needed to ensure an equal coverage of volunteer speakers . So we would need twenty ladies from Birmingham, twenty gentleman from Birmingham, twenty ladies from Glasgow, etc. The reality of what we could find in Hursley was very different, twenty ladies from Glasgow, for example, wasn't going to be possible. We did consider setting up a mobile recording unit and travelling to hotels, but there were so many practical difficulties with that idea, we decided it simply wasn't worth the hassle.

One old chap when asked his accent, he replied "Cornish" which was fine, but he thought he had to put on a Cornish farmer's accent, as if we had asked him to do some kind of weird, one-man, "audienceless" comedy set. We could hear him, even from the next door office, guffing and gaffawing, and adding a heavy "oh ahhh" to the end of each sentence, reminiscent of Worzel Gummidge. He seemed a bit disgruntled when I popped back in and gently explained that we just needed him to use his normal speaking voice, not to add any words of his own, and to start again, thanks very much. Oh well takes all sorts, I suppose.

More surprisingly we found the occasional volunteer that misread their sentences, sometimes mispronounced words, one that sticks in my mind is the word "hyperbole", which they read out as "hyperbowl", presumably an upgraded superbowl! For the most part, however, the vast majority of sentences were read correctly, and aligned with their expected phonemes perfectly. We ended up meeting or getting very close to our target of 800 speakers, and whilst the accent distribution was heavily weighted in favour of Southern and Received Pronunciation (BBC English), we did have a reasonable number of strong regional accents to help make the acoustic model work for customers anywhere in the UK.

Over the years we built two different recognition systems, we started with what we called our "discrete" system, which required users to pause briefly between each word. The discrete system went under the brand name of

“VoiceType Dictation”. In later years, around the turn of the millennium, we lifted the requirement to pause between words, this, our “continuous” system was called “ViaVoice” and worked with natural speech.

The marketing team were based in an IBM office block in Basingstoke, which incidentally, later became yet another sell-off. As well as naming the products, that team had the job of producing catchphrases, they came up with ...

“ViaVoice, you talk, it types”.

To which I cheekily replied “Something else!”.

Anyway we recorded spoken sentences with both forms of speech, i.e. discrete and continuous, which means both with and without pauses between words. This way we were able to make both forms of systems, using the different sets of recordings.

The advantage of the discrete system was that it was more accurate, i.e. produced fewer misrecognised words, the downside was that it required users to speak in this rather robotic manner, with pauses.

OK that completes the explanation of the acoustic model for which Eric was responsible. I can now move onto the final piece of the system, the language model, this was my job.

I had to start by collecting as large as possible body of text, or corpus. Clearly, I needed this as electronic data, i.e. in soft copy form, so that it could be counted and processed by program. The broadsheet newspapers had all started publishing their output as annual compact disks, so for about £100 pounds, one could buy the text from every article from say, the Daily Telegraph, for the year 1998 – conveniently stored on a standard CD. These were typically used by journalists, historians and researchers.

We had a decent budget for this project of around a few thousand pounds, so I ordered the last ten years worth of articles for each of the Telegraph, Times, Independent and Guardian. This was pretty much the entire collection in the sales catalogue.

When the CDs arrived, I immediately ran into a big problem, the newspaper articles were encrypted, to prevent copying and selling on, so whilst I could search for articles. I couldn’t copy the entire set of articles into our databases to run the statistical analysis programs to compute the word probabilities, required for recognition. I started to think that I’d spent about £4000 of IBM’s money on forty useless CDs. Oh dear.

Well I played around with the search software that came with the newspaper CDs and found that searching for a common word like “the” would return every article, but the largest range of days allowed in the search was just three days. So there was a solution, but it required over 100 searches to be entered manually to cover the entire year. As an experiment, I tried using one broadsheet CD and running the 100 searches for the word “the”, progressing through the year, three days at a time. It took me a couple of hours, but it did work. I resigned myself to a week or two of mind numbingly boring work. In reality it only ended up taking about a week, since one of the broadsheets didn’t bother with encryption, and for some of the earlier years none of the broadsheets did. So for those CDs it was just a two minute job to copy all the articles in one go. Hallelujah, for the hassle of about one week of drudgery, all forty of the CDs were usable.

I did have some other sources of data, but none came close to the size of the broadsheets, which eventually got to a grand total of 1800 million words.

The language model used sequences of three words, called trigrams. This concept had been invented years before by a well known IBM researcher called Fred Jelinek, who I did meet at a conference once. He commanded a huge amount of respect when he spoke. To explain his idea consider the phrase or trigram “bottle of beer”, if you misheard that “beer” as, say “pier”, then you would automatically correct it to “beer”, since “bottle of pier” is obviously nonsense. This is effectively the trigram language model in your head, continuously helping you to understand what might be noisy speech in a pub, or over the telephone. So our product recognition system needed huge lists of likely trigrams, and this is why we needed the millions of broadsheet articles.

You might ask why trigrams, why not use bigrams or 4-grams or even more? The answer is that bigrams result in a less accurate system, since there is less context available. 4-grams would be slightly more accurate , than trigrams but there are so many combinations of 4-grams that the language model wouldn’t fit on a CD. Trigrams seem to provide a reasonable compromise.

So next step was to create the vocabulary that our recogniser would use, and we had decided on a total of about 30,000 words, and for this first system, I simply took the most frequently occurring 30,000 words from the entire newspaper corpus.

Last step was to locate all pronunciations for each of these words. We had some soft copy, phonetic dictionaries of varying quality, but in practice I found that about 10% of my chosen 30,000 word vocabulary was missing from our dictionaries. Common placenames and people names along with unusual conjugations of verbs were missing, primarily because these words, whilst in common usage, do not usually appear in dictionaries.

So I had the problem of hand creating the phonetic spellings for about 3,000 words. The solution we came up with for this was to create 300 “sentences” each with ten of these words. I then read them out with a nice gap between each word. Next, I created a program to split the 300 sound waves up into ten single word chunks. Finally I fed these 3000 chunks, along with their spellings into a program normally used by customers to add words to their recognition systems.

The majority of words produced a sensible phonetic spelling from this program, but a few didn’t, so I ended up hand checking and correcting some manually.

So, patient reader, that is more or less it. The language model, acoustic model, recognition program and a basic word-processing package are included on a customer CD, and put into a pretty box along with the headset microphone. The price tag varied over the years, but I think we started it at £99.99, plus VAT.

Was this product a success?

Runaway success is the answer, it was top of the business software charts for almost two years, although there were separate charts for games which presumably would have been outselling us. The marketing team did a great job of pushing the software through a variety of channels including Computer and Electronics magazines, trade shows, and eventually retail outlets like Dixons, Curry’s and PCWorld. I really felt like we had made it when I spotted a ViaVoice box on the shelves of John Lewis the posh department stores. Maybe next year we really were going to be millionaires!

Decline and Fall of the European Speech Groups

The marketing team were absolutely wetting themselves for us to rush out a continuous speech general English dictation product. We had three competitors in the market, Philips, Dragon Dictate and Lernout and Hauspie. In magazine reviews we were generally best or second to Dragon.

Lernout and Hauspie were more European based, and generally low down in reviews. They got out a continuous product but the reaction was pretty poor. Accuracy was very weak by most accounts. So we did everything we could to get ViaVoice shipped. Since continuous speech is so much harder than discrete to recognise accurately everybody was looking for improvements.

I increased the vocabulary from 30,000 to 50,000, this obviously helps with coverage of words used by customers, for example it might reduce the number of missing words in a given document from 2% to 1%. The disadvantage is that there are more words to choose from and therefore get wrong.

I hand checked the most frequent 30,000 phonetic spellings, since I kept noticing recognition errors caused by missing or incorrect dictionary entries. This took me a couple of months, but probably avoided about 1% of dictated errors.

On the acoustic side, the US guys came up with a scheme to produce a ranked list of phonemes, rather than just the top choice. Eric rebuilt our acoustic model based upon this new ranked based system.

We also switched to a higher frequency sampling system, to make the sound a little clearer.

Marketing had received quite a lot of complaints about the microphone being cheap and flimsy, so they increased the budget on that and found a much improved model.

We increased the number of training sentences, that customers read aloud before being able to dictate. These training sentences help tune the acoustic model to a customer's accent. We referred to this as speaker dependent recognition. These training sentences were also designed to explain to users how to use the system properly, and since they were a pre-requisite, no one could say that they hadn't read the instructions!

Well even with all these improvements taken together, our first release of ViaVoice was still less accurate than the discrete VoiceType Dictation. I wanted more time but marketing insisted on getting it out to customers. Well, the reaction was pretty intense, but of course loads of customers were not happy with the accuracy, and phoned up to complain. In fact so many phoned up that we had created a new job for many Greenock employees, they set up a call centre to deal with them all!

All the European teams were operating in parallel, so we regularly met to discuss ideas and to help each other out. We usually met in Heidelberg, but occasionally met in Paris, even visiting the Eiffel tower once. I'll explain a little about each of the teams...

The German team was in charge, and led by a senior guy called Seigfried, who for some reason was known as Jimmy. He was one of the original members, and was the first to create a German recognition system. Martin did the German language model, he spoke particularly perfect English. The main problem with German is the vocabulary, since they glue words together,

resulting in needing about a million words, which was obviously impractical. They had some tricks to join up words, but it wasn't entirely satisfactory. Another issue they had to deal with was that the German government mandated a new spelling system for the entire country part way through our project.

The French team consisted of Hubert, Claire and Jean-Christophe. French was probably the hardest language, they are plagued with homophones, words with the same sound, but different spellings. They also had to deal with liaison and elision, which cause phonetic variations between words.

Spanish is probably the easiest, since they have few homophones, plus their spelling to sound rules are generally straightforward. It wasn't unusual for Spanish speakers to get a 2% dictation error rate. Jorge did the acoustic system, Paco the language model. Paco could speak all the languages!

Finally, Giulio and Marco did the Italian system, this was about second easiest after Spanish. Giulio was the brains behind the class language model, based upon clusters of words. The whole team produced a scientific paper demonstrating that recognition accuracy improved regardless of language, by adding a class language model.

So all the teams worked together, and communication between us was entirely in English, since that, as Eric used to say was the lingua franca of Europe. The real reason was that none of the English speakers were proficient in anything else. We really were a happy family working as a phenomenal team, having weekly conference calls and occasional meetings.

We released a new version of ViaVoice to line up with Christmas every year. I can remember at least three called...

* "ViaVoice"

* "ViaVoice Gold"

* "ViaVoice Millennium"

The last of which must have been the year 2000. We were no longer top of the business software charts.

Each year the newer product made improvements in accuracy, but of course very few customers would pay out every year for another try, particularly if their previous experiences had been bad. The Greenock help line had to deal with all the different versions and languages, which couldn't have been easy.

The final UK release had a vocabulary of 150,000 words which required a huge amount of work to create all the phonetic spellings. Whether it was worth the effort involved, I very much doubt.

Our competitors were not doing that well either, I think Philips quietly withdrew their product. Lernout and Hauspie got tangled up in a legal dispute, relating to over optimistic venture capital. I believe that Dragon managed to hobble on, albeit with redundancies, and I think they are still going even today.

We also hosted any other country interested in building a recogniser. Egypt did one for Arabic, although the guy that ran that was really part of the European Speech teams.

Scandinavia sent about six speech experts to Hursley, they were here for a couple of months, so a bit long for a hotel, but short for a rental. We eventually found them what turned out to be student accommodation on the Winnall estate, hardly luxurious, as they politely observed to us. They could all speak several languages, one girl could do British Accents as her party trick. They did recognition systems for Swedish, Finnish and Norwegian if I remember correctly.

We also hosted a couple of people from India, to do Indian English, which amongst other minor differences had a quite different system for numbers, they don't use thousands and millions, instead they have lakhs and crores. A lakh is 100,000 and a crore is 10,000,000.

We also did an Australian English, but I think that was all done via conference calls and emails, rather than hosting anyone from down under.

Chinese was thought to be a large market, not only because of the population size, but also the difficulty of use of keyboards with hundreds of keys. There were many difficulties with Chinese, not the least of which was finding word boundaries in their corpora, since they don't have a space character. The US hosted some Chinese folks to deal with all that.

Quite often I was involved in helping with all these other languages. Obviously I speak none of them, but when the speech engine crashed, I could often track down the reason, and save people time who were working on their own languages. In a strange sort of way you could say that the speech engine understood all the languages, and could be used as a kind of universal babel fish.

So the competition forced everyone to lower prices, our ViaVoice product hit £39.99 at which point we no longer making a profit.

We branched out into other areas. Telephony seemed to be a hopeful prospect since we would be selling to enterprises rather than individuals.It was around the time that automated call handling was taking hold, these were the annoying “press one for sales, two for complaints, etc.”.

I worked on what we called the IBM Voice Dialler. This was an automated telephone operator. A typical call went as follows:

SYSTEM: Hello, please state the first name and last name of the person you wish to call.

USER: Bob Grainger.

SYSTEM: Do you want to be put through to ... Bob Grainger?

USER: Yes.

At this point the system redirects the caller to me.

This seemed a reasonable time saving facility for companies of a few hundred or thousand employees.

It all seems fairly straightforward, but was anything but. It required a speech recogniser, a speech synthesiser, plus telephony hardware. It also needed easy to use interfaces to keep the company directory updated as people joined and left. The whole thing was a nightmare of complexity, so much so that the installer package needed an installer itself! I built up the first system in Europe, and for a while it was a popular way to call people in Hursley.

Although it worked quite well, I don’t think we ever worked out a way to make it profitable.

We also worked on recognition in cars, this required recording of speech samples whilst driving along, to build the acoustic model. We hired a driver and recorded a few people, but it always felt dangerous to me even with three people in the car. So instead we recorded in a stationary car, with the intention of mixing in engine and road noise later on. My car was used quite frequently, at least up until the time, one evening I left for work to find a flat battery, and had to phone my Dad to rescue me by starting my car with jumper cables to his car battery. After that my keys stayed firmly in my pocket.

I don’t think we had a lot of business success with in-car speech recognition either.

The writing was on the wall, speech wasn’t making enough money to sustain all the people involved. Even the marketing manager from Basingstoke phoned me to advise me to look for other roles. So, with a heavy heart, that is

what I did. Since Storage Products had recently moved from Havant back to Hursley, that seemed the obvious choice. I could view my seven year speech job as a “Sabbatical from Storage”, and I was returning like the prodigal son.

I was one of the first to leave, but one by one everyone else left. The only people to remain were the US researchers, since they are funded by a flat 6% of IBM revenue, and since they are doing research, product profits are not a requirement for them. They, some 17 years later are still producing academic papers demonstrating improvements in recognition accuracy, and today (2017) are claiming close to human performance, with most of the improvements coming from brain-like computer systems known as neural nets. Quite remarkable strides.

Eric was meanly offered a demotion to some job he wasn’t in the slightest bit interested in, and since the pension scheme was being hacked back, he choose instead to take early retirement and protect his defined benefit pension scheme.

Seigfried, the German and European leader also left IBM to work for a company called the European Media Lab.

There was some arrangement to maintain copies of all our recordings and text corpora, but the chance of being able to revive it now seems slight.

So a somewhat bitter sweet ending to my speech days, I packed up my stuff and moved blocks back to Storage.

#pl_scsi

So I was back in Storage again, they needed me so much that they relocated everybody back from Havant to Hursley again! This a complete reversal of the decision made about eight years earlier. It was summer of 2008, and our son, Tom had cadged another work experience vacation in our department.

Question was what to do with him. Well my pal Nick had an idea for our virtualization storage product, known as SVC standing for SAN Virtualization Controller. Where the SAN stands for storage area network. What Nick wanted was a test facility to allow arbitrary commands to be sent to disk and controllers.

What are these arbitrary commands? They are any SCSI commands, where SCSI stands for Small Computer System Interface. For example, reading and writing blocks on a disk, starting a motor, loading fresh code. SCSI is pronounced “scuzzy”

I said that it looked a bit of a tough challenge, however, there was an existing test module that would make a perfect template, from which Tom could start.

Why was it called pl_scsi Well every module in our system has a two letter prefix, and the pl prefix stands for platform layer.

Having got all the alphabet soup clarified, I can press on with the story.

So I set up all the necessary access rights so that Tom could edit his modules and check them into our system. I pointed him at the template as the starting point to copy and paste from and left him to his own devices.

Well he plodded through this job for three or four weeks, occasionally asking questions, but seemed to be making good progress. Sometimes his questions required help from engineers other than me, this helped get Tom known around the department, and he wasn’t shy about talking to them.

It seemed like he was running out of time, since we had some vacation time booked, and there were only a few days left. Debugging we knew was going to be tough, since it was one of those programs that has to work in its entirety or not at all.

The best way to debug it was to use an instrument called a protocol analyser, these units cost about £100,000 each, mainly because the market for them is specialised and small, so most of the cost is in their development. The analyser was made by a company called Finisar. Just learning how to use these instruments can take months, fortunately I was familiar with it so could help Tom. Although he soon got the hang of it, just like any of the tech stuff he had grown up around.

The process of debugging followed these steps:

1. Compile the latest version of pl_scsi

2. Download it to the SVC system hooked up to the Finisar Analyser.

3. Walk down to the lab with the SVC and analyser.

4. Set the analyser capturing fibre channel link data.

5. Run pl_scsi with a SCSI command.

6. Stop the analyser.

7. Pour through the analyser trace to see what happened, or didn't happen!

8. Figure out the bug, walk back to the office.

9. Fix pl_scsi, and go to step 1 again.

Well we went around these steps several times gradually getting further and further until finally at 7pm on Tom's last day it worked properly, we could send any command we wanted to any controller or drive. Oh the power!

Tom was pleased to be done, we went on our holidays, and he and I later returned to school and work respectively.

Tom's code needed checking in to our version control system, so I did that, and then everyone could start using it. Boy, did they start using it, people from Germany, China and the US all wanted to do things with it. What we found was that it had all sorts of uses that we hadn't anticipated. It could workaround and fix unrelated things like vital product data errors. It could be used for all sorts of unexpected testing and debugging.

Later I began to realise that I could build new tools around the pl_scsi infrastructure, such as downloaders, enclosure command line interfaces and even an automated media error injection tool.

It wasn't long before other companies wanted pl_scsi too. Initially we were reluctant to do this, because in the wrong hands all sorts of trouble could be caused, such as wiping a disk full of data with a single SCSI Format command. We solved it by only allowing installation if the machine serial number matched a predefined list, this enabled us to control access to pl_scsi.

I did find one easy-to-fix bug a few weeks after Tom had finished it, but other than that no further bugs were ever found. For some years after, I had to deal with a continual drip, drip of emails asking for help, or claims of bugs, or new feature requests. This was mostly a kind of proud-dad pleasure.

Tom was doing his A levels, pretty much unaware of quite what he had achieved. The moral of this tale is that kids rise to a challenge, don't underestimate them, they will probably surprise you!

To infinity and beyond

So we are at around 2015 and my role in storage was changing, or evolving into a consultant, predominantly for disk drive qualification. My nominal position was as a tester.

For each different type of drive to be used in our systems, an extensive series of tests must be performed to ensure that each drive will operate correctly and perform to specifications. We call this “drive qual” for short.

Since the drives are made by only a few vendors, we have to liaise with each vendor to ensure this process works smoothly. We used a weekly conference call between IBM and each vendor.

The vendors are, of course, in competition with each other, so we have to be careful not to communicate between vendors, and also to ensure that we treat them fairly. Using multiple drive vendors is a big advantage for IBM and its customers since we can use the concept of second sourcing, which basically means if we get issues with drives from one vendor, we can switch to an alternative vendor. This works out well for drives failing in the field.

Drives come in many varied forms, there are two physical sizes (2.5 and 3.5 inches). There are mechanical, spinning platter drives and solid state drives with no moving parts. The spinning drives come in different rotational speeds. Finally, the capacity, i.e. how much can be stored, also varies dramatically.

All these differences mean that we end up with over 100 different drive types all requiring qualification, and field support.

My role in this started when my manager asked me to join in a Seagate Technologies conference call to assist with discussions and troubleshooting some drives. He asked me because he was aware of my previous experience in IBM when we developed and manufactured our own drives. A lot of this experience was 20 years old, but I hadn’t forgotten much, and a lot of the principles had remained the same.

Once I had contributed to the Seagate call, then I got invited to the Hitachi call, and a little later, the Toshiba call. Several times I was able to troubleshoot an interesting problem that we ran into, just by chatting on the phone to the experts. Even though I never met most of the people on the calls, after a year or two of regular weekly calls, I felt that I had got to know them and even considered them friends.

I helped with anything and everything related to drives, including early qualification which we often call “bring up”, through performance measurements to field support. Field support is a fancy term for helping resolve customer complaints.

The biggest open question in the disk drive world is whether mechanical spinning platter drives will be made obsolete by solid state drives In some fields such as cameras, smartphones and tablet computers they already have, simply by being smaller and lighter. Newer laptops and PCs also appear to be phasing over to solid state . The real question is what about enterprise

storage? The problem there boils down to cost. A mechanical spinning drive of similar capacity is about three times cheaper, at the time of writing (2017), and it isn't clear to me that the price difference is going to achieve parity, i.e. match. There is also a supply and demand problem, since the world market for storage capacity could not currently be satisfied by solid state drives because there are simply not enough chips made each year.

For years solid state drives were considerably smaller in terms of capacity, and certainly the world record capacity had always been held by a spinning drive, but then, in 2016, Tom, my son, pointed me at an article stating that Samsung had come along and rewritten the record books with a jump to a 15TB in a single drive. To cap it all they had not only broken the previous record by 50%, they had the audacity to fit it in the smaller 2.5" form factor drive! The only slight issue was cost which hadn't been announced at that time but was thought to be around $10,000. This seems like a ridiculous amount for a single drive, but given that it could easily replace fifteen 1TB spinning drives, it wasn't too bad.

Managers at Hursley also noticed this record and very quickly they wanted to add this drive to our product portfolio. The only trouble was we hadn't had any dealings with Samsung before, so we had to set up new contracts and agreements between them and IBM.

I was asked to help with this work, and I was more than keen to work on this, the world record drive.

I was lucky enough to get a very early model of the world record 15TB drive on my test stand. First thing I did was check that it really, really had 15,000,000,000,000 characters of storage! Yes it did!

The qualification of this drive went relatively smoothly, any minor issues that we did find were addressed by Samsung, and it wasn't long before the drives were available to IBM customers. I was quite proud to have helped with a world record drive, not for the first time in my career, but the last time was about thirty years earlier.

So the question was how had Samsung managed to steal a march on the other drive vendors? I don't think they have any "magic secret sauce", rather they have managed to develop their chip technology a little earlier, and they have managed to make best use of the third dimension.

What is the third dimension? Well since the first integrated circuits were developed back in the 1960s, they have been made on planar chips. This means that the circuits are etched in a flat array, i.e. a grid. Only recently have chip manufacturers started to etch circuits in layers on top of each other. So, now the dies have layers of grids of memory cells and the chips have stacked

dies and the drives have stacked cards with multiple chips. Even the individual memory cells have multiple bits of information stored in them.

The other drive vendors are already doing much the same things, but it is a continual race, everyone wants to cram as much storage into a small a space as possible. Meanwhile world's appetite for more and more storage shows no sign of abating.

In a similar, but different, direction, our virtualization product has been looking for gains by cramming more drives into each drawer. For many years, IBM storage had an implicit rule that prevents the movement of one drive when replacing another. The idea behind this rule is common sense, namely don't disturb any working drives when replacing a failed one. This rule has generally implied that only one or two rows of drives are incorporated into each drawer of drives.

Well to get to the ultimate packing density this year (2017) our storage products department dispensed with this rule, so managing to fit 92 large drives in a single drawer. It is still possible to replace a drive without powering the drawer down, but of course the drawer full of operational drives does have to be gently slid out. The drawer when opened up just contains wall-to-wall drives. No space left!

So why do I mention this? Well in terms of storage capacity, we really seem to be approaching the end, in some sense you could say that our job is done. Of course drive vendors will be building even more capacious drives, but they will be yet more expensive. We have run out of dimensions! There is simply no more space left to cram more storage!

We've hit the limit.

THE END

This is Bob Grainger's story. His autobiography communicates the influence of his father on his thinking, their shared fascination with maths and building things, and his career with IBM where problem solving and debugging were his daily delight.

He reflects on his youth in Birmingham building a computer from a kit in his bedroom and his failed shot at being entrepreneurial, which almost got his uncle sacked from his job.

The later chapters chart Bob's career with IBM. Bob's narrative throws a refreshing light on a company regarded by many as being too tough on its workforce. His joy in problem solving flows through his writing.