Professor Harold Thimbleby

Interviewed by

Elisabetta Mori

16th January 2023

In Cardiff

Copyright

Archives of IT

(Registered Charity 1164198)

Welcome to the Archives of Information Technology. It's 16th January 2023 and we are in Cardiff. I am Elisabetta Mori, an interviewer with the Archives of IT. Today I'll be talking to Professor Harold Thimbleby. Professor Harold Thimbleby is See Change Digital Health Fellow at Swansea University, Wales. His research interests span from human computer interaction to formal methods and their medical applications. His passion is designing dependable systems to accommodate human error, especially in healthcare. He is an Honorary Fellow of the Royal College of Physicians, a Fellow of the Royal College of Physicians in Edinburgh and a Fellow of the Learned Society of Wales. He is also an Honorary Fellow of the Royal Society of Arts. Harold has been Royal Society Wolfson Research Merit Award holder. Finally, he is also Emeritus Gresham Professor of Geometry. Welcome Harold.

Hi! Thank you very much for that lovely introduction.

Thank you. So let's start with a very, very easy question.

Okay.

Where and when were you born?

I was born in a little town called Harborough Magna near Rugby in Warwickshire and I was born in 1955.

Can you describe your family, your parents?

Yeah. My mother was a musician, but she became a fulltime mother, and my dad was an architect and we moved to Rugby with him to have a practice as an architect which he, you know, he designed all sorts of things from people's extensions to factories, to whole rows of shops and stuff like that. I remember working in his office a lot. He was the son, obviously, of my grandfather who was an electrical engineer and I spent most Saturdays round at my granddad's and he had a workshop and he just got me excited with electronics and engineering and valves and sparks and stuff like that.

Thank you. Can you describe your family life as a child?

I grew up with my mother’s parents, the other set of grandparents, and they had a big house and one end they lived and one end we lived. And my dad was an architect, he then built an extension which was a sort of general purpose room and had a row of, a bench along the end with cupboards, and my sister had some, we both had some, and I had electronics at the other end. So I gradually built a little lab, I built an oscilloscope and stuff like this and did electronics and I’ve never really thought about it, but I expect my parents and my grandparents had a chat, you know, this was getting too much for them, and I was offered a room at the other end of the house which became my lab. And I’ve sort of realised recently that that’s sort of what changed my life. I spent all my time in that room, building things and blowing things up and testing things and doing sums to design circuits, and I had a lot of fun. And I’ve realised people get good at things when they practise, you know, become a good footballer or a good dancer or something, and I got really good at electronics. And at school it was A level qualifications and I did physics and I came top in the country, because I was so into doing electronics and physics at home, I’d been doing it for like ten years by the time I got the national exams. And I then went to university to read physics, seemed the obvious thing to do. Although I’d played with the computers, the local college in Rugby had a computer I’d been using, but I didn’t think of computers as a serious subject so I did physics. And at university, you know, the first year laboratory for doing physics experiments had a PDP-7 in it, and I started playing with the PDP-7 and I’ve never looked back, I’ve been a computer scientist ever since, really.

[00:04:44]

So was that the first computer you saw in your life?

No, no. First computer I got my fingers on was an Elliot 803 at the local Polytechnic.

When was that?

Gosh. That would have been the sixties sometime, because my uncle, who lived in Rugby as well, was a lecturer at the Polytechnic. And, you know, he had been an engineer during the war so he, you know, he supported me in saying, ‘Come and have a look!’ And so I, I, I, was I about ten or eleven when I got my fingers on the 803 and programmed it a bit. And then the Polytechnic got access to an ICL 1900 and I used to cycle there and drop off cards for the 1900 to run programs. So I had a lot of fun.

What were you programming on that computer?

I tried everything. ALGOL 60 is where I started, then I learnt Fortran, BASIC. I discovered if I ran BASIC programs I could get more time on this mainframe than if I ran ALGOL 60 programs. So ALGOL 60's a much nicer language to program in, you can have more fun programming. But I as a mere, you know, teenager, I could only get sixty seconds and in BASIC I could get, I think it was five minutes. So that sort of transformed what I could do. And then I came across ML/1, which is a language by Peter Brown at Calgary, and started doing that, and that's a micro-processing language, so it's completely different. And then my school got a teletype and a modem and linked up to the Modular One computer at the University of Warwick and that I started programming in BCPL, learnt BCPL, then got in touch with people like Martin Richards who'd designed BCPL, and I did an A level in computing in my spare time. But as I say, I didn't think of it as a real subject, it was fun, but physics was a real subject. So that's why I did a degree in physics.

So what do you remember of your undergraduate years in physics?

In physics? The first year physics lab, where the PDP-7 was, did something very clever that's stayed with me and it's been really useful for when I teach, is there were perhaps, ooh, twenty-five experiments set up round the room and us students, we went round the experiments all together. So every week you did this experiment and then you moved on to do that experiment, and so on. And the job was to study cross-polarisers and draw graphs and things and just get into experimental methods. But every experiment had got something wrong with it. So the cross-polarisers were about half a degree out, so when you did the experiments you didn't get the result the theory said because you're half… And the real physics was to sort out the errors, rather than to do the sort of textbook experiment. So when I later became a lecturer in computer science, students would complain that the EPROM programmer didn't work or something, and I would say, welcome to the real world, you know. It was learn how to use it, that’s what the important part of computer science, not having perfect equipment.

So you spent three years studying physics?

Physics, yeah, yeah.

Where did you study physics?

Ah, a place that no longer exists. It’s Queen Elizabeth College in the University of London, and that was taken over. And I did some of my degree at Imperial College, which was just about a mile away from Queen Elizabeth College, and did neuro-physics and stuff like that. And then in the… I was going to say in the holidays, I got jobs doing programming, but actually, during termtime I started running the College’s access to London University’s mainframes which are large CDC 7600s and 6600 computers. So each college had a Modular One, in fact, which I’d been using at Warwick earlier, so I knew my way around it, and that would process people’s card decks and submit them and print out results. So, I had fun doing that. And one holiday I became the chief programmer at the Royal College of Art. They had an Elliot 905 with a vector graphics radar display, and they had a fixed head drive of 100 megabytes, which was, seemed like a big disk. I know it’s tiny nowadays, but it was great fun and artists would come in and draw fun vector graphic things and there was a light pen, so it was interactive graphics, whereas most of the work I’d done previously was card decks and you sent off something and you’d get the answer back later. But interactive programming was good fun.

[00:10:04]

Did you develop friendships when you were at the Royal College of Art, with artists?

[laughs] No, sadly. I got to know the people who were working in the computing team, but the computers were down in the basement, so keeping them working, we stayed in the basement, basically, yeah.

So what was the relationship with the artists like? You were helping them program…

Yeah, they would come down with a project and work on it. And I've – gosh, it's a long time ago now – so I've just remembered, this was in the Department of Design Research, so there were lots of designers, you know, drawing structures and things and it was fun.

And then what happened?

Then I did a Masters degree in computer science at Queen Mary College. And the notable thing there was Queen Mary was the first place out of America that got Unix. So the lab I did my PhD in had got Unix and that was great fun. And the laboratory's sort of research theme was the office of the future, now how are we going to design computers to run the office of the future. They were all excited about window displays and colour displays and mice and things like that, and it was going to change everything. And so we were way ahead of the game. But as a physicist in my sort of back of my mind, I was thinking, what evidence have you got, what are the theories that make you believe reliably that it is going to be the office of the future. I can see it's fun and things like Ataris came out somewhere round about then, so you know, interactive computing started to hit the mainframe and… But there was no argument that I could see that this was actually going to change offices, other than excitement. So I did my PhD in what we now call human-computer interaction, about the theories of computing that made some sort of sense about why this will work for interaction.

So you did your Master thesis in 1977 and your PhD in 1981.

That's right, yeah.

And both of them at London University.

Yeah, at Queen Mary College. Yes.

Now, in '81 it's a moment in time where human-computer interaction starts to have…

Started to get off the ground, yes.

Exactly. So, can you describe, from your perspective, what were the big changes that led to this?

Mm-hm. That's a good question. Doing my PhD I had – well, in fact, doing my Masters degree – I'd started to get in touch with the computer scientists who worked in what we now call user interface problems, because just getting the computer to work at all was the big problem, so it was an engineering problem and - I'll say everybody, but it wasn't quite everybody - everybody I talked to was a computer scientist or a physicist who'd turned into a computer scientist and they were trying to get this stuff to work. And gradually I started to come across people like Brian Shackel, who was actually an ergonomist, and he'd got into it the other way. And so, as I did my PhD, the whole world went from what we now call HCI, we would have said, if we'd had the words, we would have said HCI is computer science at the beginning of my PhD, and at the end of my PhD we would start to say, well, HCI is psychology and computer science. And then a couple of years later the Alvey project got off the ground and then people started to say, well, HCI is psychology. So it shifted over that period quite a lot.

[00:14:02]

During the years of your PhD, did you work with other colleagues on the same theme or were you doing these on your own? So were you part already of a community?

Right. There wasn't an HCI community, but it was starting to build up. So I came across Brian Gaines, who was the editor of a journal called the *International Journal of Man-Machine Studies* – IJMMS – and nowadays you wouldn't say 'man-machine studies' and it's become… I've forgotten what it's called now, the *International Journal in Human-Computer Interaction*, probably, I've lost track of it. But it's, you know, MMI – man-machine interface – obviously has become human-computer interaction now, because everybody uses computers, not just men. [laughs] But, yeah, when I started people didn't think of that, it was out of our consciousness. We weren't, just getting it to work at all was the problem. So there's Brian Gaines was a leader and a lot of what I was doing in my sort of day-to-day research was involving building text editors and editing and how do we type. And I discovered John Long at UCL, he was an ergonomist and he'd done a lot of research in typing, so I went over, met him lots of times and learnt a lot about the psychology of typing and how we correct typos and things like that. It was fascinating stuff. But you need the editor to support that sort of thing.

So what do you think were your major contributions of your PhD?

The major contribution was the work I did with Richard Bornat. He turned up as a new lecturer and he wanted to build an editor. So he wanted to build an editor and I was looking at interaction for typing and – this is on a time-sharing computer running Unix that was struggling doing anything with the lab of maybe thirty people. It was quite a big lab, so the PDP-11 supporting that number of people time-shared, you have to program it very carefully. And Richard and I came up with some really good principles of word processor design, which I think make Pages and Microsoft Word look primitive. One of the things is we decided you're editing a picture, you're not editing text. So you've heard of What You See Is What You Get? So, what you see on the screen is what you're editing, whereas if you use Pages or Word or anybody else's editor, you've got to remember, if I point on your screen, it'll make sense to you, but people listening to this will hear it. But you've got a line of text there and after the end of the line of the text, what happens if you type there? There's nothing there. Whereas in our editor it was all, it was as if the screen was full of spaces, you can type anywhere, it's all text, you can delete things anywhere, you can type beyond the end of the document because what you can see is what you've got. Whereas in Microsoft Word it's got this really quirky idea of what a tab is. Is a tab, you know, a gap with spaces in it, or is it tab, and you have to remember, you know, here's a piece of text that's indented, did you indent it with spaces or with a tab? It matters in Word. Whereas in our editor, are all vanished. What you can see is exactly what you've got.

And so after your PhD, what happened to your career?

I had a year lecturing at Queen Mary College and then I had to find a job. And I got offered two jobs. Barts Hospital wanted me to set up their digital systems, get Unix there, and York University offered me a lectureship. And by that stage I'd got two toddlers and having them in London, you sort of think, well actually, we'd all have a better quality of life if we moved to York. So we went to York.

So when you say 'we', you're talking about your wife?

Me and my wife, yeah, and the kids.

So when did you meet her?

Well, during my PhD. So 1979, I suppose. Yeah. Yeah, I met her and we got married, got married in '80. And then we had a race, are we going to have a baby first or a PhD thesis, and she won. So [laughs], I was still struggling to finish my PhD. [laughs]

[00:18:48]

And then, so you moved to York.

Yeah.

Computer science faculty. What do you remember of those years?

University of York computer science then was remembered for its work in Ada and it had some VAX-780s and some PDP-11s and I ended up being the guy in charge of some of the Unix systems. So I was sort of in the core of the computing of the department, but my interests were user-interface design or man-machine interaction or HCI, as it was becoming known. I think the first British Computer Society HCI conference was probably at York. One of the earliest British Computer Society HCI conferences was at York because when I went there, some of the psychologists, Andrew Monk and Nick Hammond, knew about me and we started having weekly meetings talking about the problems of HCI and they had psychology and science and Colin Runciman and Michael Harrison, who were close colleagues of mine in the computer science department, we realised, this is a thing we can do jointly.

So there was a nice community of HCI in York?

Yeah.

Were there other communities in the UK at the time?

Hm, that's a good question. I'd be tempted to say no. There were lots of individuals, but we had a good critical mass. There were places like the Applied Psychology Unit in Cambridge, which as its name suggests, it was full of psychologists who were keen on using computers and understanding interaction from a psychology point of view. There was the ergonomics group at Loughborough, there was the ergonomics group at UCL. Peter Johnson was somewhere. He definitely ended up in Bath, but it's a long time ago, I don't remember where he was then. There's Alistair Sutcliffe. There was quite a community of computer graphics, which was Computer Science Department, and there were people who bridged computer graphics with HCI. I remember Alistair Kilgour at Heriot-Watt University. Steve Draper at Glasgow. Lots of individuals, but I don't remember many groups. So we had quite a big group at York.

And how long did you stay at York?

I think six years. And I got offered a Chair of Computer Science at Stirling University. And [laughs] one of the – I'll call it a mistake – one of the mistakes I made is Peter Henderson worked at Stirling University and I thought it would be great to work with him, and I didn't realise until they offered me the job that actually the job was Peter Henderson's Chair, he had moved to Southampton. [laughs]

During the years at York, you got grants, thanks to the Alvey project.

Yeah, yeah.

So what was your project specifically about?

Well, we called it the Five Man Project, because it had got Nick Hammond, Andrew Monk, Michael Harrison, Colin Runciman and Harold Thimbleby, me. So five men on this multidisciplinary project. I would like to say it was based on a paper I wrote in my PhD about generative design principles. So there was a large literature as computers got off the ground on design rules, guidelines. So Sid Smith at Mitre had an enormous book of, you know, things like the delete key must delete characters or something, and you must have undo. And these are all sort of ad hoc, and we realised if you had a theory of them, then if you formalise them, which is, you know, I'd got into formal methods, if you formalise these principles you can start saying well, do they work together. So one of the things we discovered is undo was really quite an interesting principle. You think it's kind of obvious, if you make a mistake you want to undo it, fine. Well, when you use formal methods you start thinking, well, what happens if you undo the undo, now what does that mean. And it turns out you've got to make some decisions about what it means, but if you just say your computer system must have undo and you haven't thought about these decisions, you're going to end up with bugs, or unexpected properties.

What year did you leave York?

I would think '88, and then I went to Stirling. Yeah. And then I left Stirling in '94 and went to Middlesex.

Middlesex University?

Mm-hm.

Was already not a polytechnic any more?

Correct. It was a university when I got there, yeah. And I went to Middlesex University to help set up the Computer Science Department. Norman Revell got appointed at the same time, he came from City. And the University originally was one Chair and they interviewed about ten of us over several days. It was quite interesting, we got to know each other and people started scheming and some people dropped out, you know, somebody said, well, we can't get the job when you're applying for it. And Norman and I carried on and at the end they, they asked Norman who would you like to work with, and he said Harold. And they asked me who would I like to work with, and I said Norman. So they appointed us both and said run the department between yourselves. And we then started scheming and Norman took charge of all of the teaching and administrative side and I took charge of all of the research side. So we built the department up from a, I suppose, six people hanging around – that's a rude way of putting it – but there wasn't really a department of computer science when we took over, and in a few years we had twenty and then, you know, it grew very quickly as we got it going.

[00:25:10]

So we are in the mid-nineties since you started working in computers, roughly twenty years passed, a new discipline called human-computer interaction was established. Society changed and computers…

Yeah, and mobile phones came out, the worldwide web came out. So, yeah, things changed very quickly.

So what do you think were the main changes and the way HCI contributed to them?

I think industry and universities are fairly separate things. They've got different goals and attitudes. So we did some really good work at Middlesex: Gary Marsden, Matt Jones, me, on designing mobile phones. And we got several patents because we got some cute ideas. Nobody was interested. You know, we talked to Motorola and Nokia and so on, and we'd have meetings with them and they would agree that what we were doing was better, and yet, you know, they want to make money, they want to sell things and thinking like this wasn't how they thought. It's a shame. So I think until Apple came out with the iPhone a lot of the mobile phones we had were very sub-optimal and confusing and, you know, the group of us, we wrote lots of papers analysing these things in different ways and you discover lots of ways of improving them.

Until when did you stay at Middlesex University?

2000, I would think, 2001. So John Long was at UCL and he retired, which left the Ergonomics Unit leaderless, in principle, so they advertised for somebody to take it over. And I put in a bid saying HCI is psychology and computer science, let's have UCL Interaction Centre, UCLIC, as I called it, which brings psychology and computer science together to look at these sorts of issues, design of mobile phones, anything that's interactive and has got a computer in it. And they bit the bullet and appointed me. Gosh, that was just before the bombing on the Trade Centers. Because I did a trip round lots of American universities, starting the day after the Trade Centers were bombed, and it was a very interesting experience.

What was your working experience abroad? Did you travel a lot, did you have a lot of contacts?

Yeah. I spent about a year at the ETH in Zurich during my PhD at Queen Mary. They were, Nik Wirth, who'd developed Modular Two, and PASCAL for that matter, he was developing a piece of hardware called Lilith, and I worked on building the text editor for Lilith, which was a, basically a Modular Two rip-off of what Richard Bornat and I had done at Queen Mary College. And the lab I was working in was all involved with the hardware and the serious programming language issues and implementing compilers and things. And they were pretty disdainful of user-interface problems. But they liked the editor. [laughs] So that was ETH in Zurich. I spent three months at the University of Cape Town with one of my PhD students, Gary Marsden, he then ended up as a professor in Cape Town. Spent quite a bit of time in New Zealand. In fact I've lectured at every university in New Zealand, which was a fun trip.

What year was that? If you remember. Roughly.

'98, possibly.

[00:29:30]

Yeah. I took my son, Isaac, with me to New Zealand. Prue, my wife, and I had an agreement that if I ever went anywhere for longer than a weekend I would take a child. So I took Isaac to every university in New Zealand.

How old was he at the time?

Well, I'm just thinking, he was about ten. So… yeah, he was about ten, he had fun. And then I had some sabbaticals in the University of Calgary in Canada and I chose University of Calgary because that was where Brian Gaines had ended up as a professor. And met Ian Witten, and Ian Witten moved to New Zealand and two of my PhD students ended up working in New Zealand. And my colleague, Matt Jones, who I said did some work with at Middlesex University, he ended up… University of Waikato, and in fact it was him who organised the trip round New Zealand. So quite a lot of my life has ended up going to New Zealand, and it's a lovely place and I was offered jobs there and I was very tempted, but it's actually a long way away, so didn't go.

What was your work environment when you were at UCL, when you established the UCLIC? What do you remember of those years?

We had a corridor in the psychology building, basically. And we had offices and a laboratory and a little lecture theatre and seminar room and, you know, the usual stuff to do everything you need. So that was fun.

And then what happened?

[laughs] When I was appointed, psychology and computer science were very keen on this and I turned up and here was the space. But when I turned up to start the job, both heads of department had changed and psychology and computer science were in different faculties and they were in different divisions of the University, so this corridor, they both wanted to have it. So quite a lot of my time was spent politicking, you know. One day I came in and the locks had been changed, in a sort of… it was

crazy. So I got the University to move us out of psychology building and we moved into a place called Remax House where we had our space and we thrived, it went really well. But, ah, I got fed up politicking rather than doing science and interaction and so on, it seemed like a waste of my life and I started looking around for other jobs and spotted Swansea University. In fact, Matt Jones spotted it and he said why don't you apply to Swansea, and that's how I ended up in Swansea. And when I got a Chair in Swansea, Matt Jones came and joined me and we set up the FIT Lab, Future Interaction Technology Lab.

Which is the HCI lab in Swansea?

That's right, yes.

Can you describe the FIT Lab, has it evolved from when you founded it to now?

Yeah, yeah. It's… well, I'm now an Emeritus Professor so I haven't kept up, and of course lockdown and so on, the world's got complicated and I've moved to Cardiff as I've retired. So I'd remember how it was, I can't speak to it today.

Because when you established it, it was 2005?

Five, correct. Yeah, yeah. And it was very similar to UCL, we more or less had a corridor in the building with offices and a lab at the end where the PhD students could work and we can have seminars and build things and do work together. Yeah, it was fun.

Slightly afterwards, in 2007 you published your book Press On. *It was published by MIT, winner of the Computer and Information Sciences Award from the Association of American Publishers. So that had success?*

It did. I was trying to say interaction, if I took an arrogant position I would say before you start studying the psychology, you've got to get the engineering right, because otherwise the psychology insights you get is an artefact of the cock-up you've made in the engineering. So the book*, Press On,* is all about how to design interactive systems, to program them so they do what you want them to do, and you can analyse them and do all sorts of things to optimise them. If you want to reduce the number of keystrokes to do tasks, start with the engineering and then you can do studies to see whether that, you know, people like it, which is not an engineering question. But that's interesting psychology.

[00:34:42]

If we jump to 2021, you publish another successful book, that is Fix IT: See and Solve the Problems of Digital Healthcare. *So when did you get involved with digital healthcare?*

Ah, that's a good question. When I was at UCL one of the Masters students got run over and I visited him, he was in intensive care, he had a pretty bad accident. He was a keen motorcyclist, I think a car knocked him off and drove over him so he was in a mess. And I was just shocked at the equipment he was connected to. There was one infusion pump giving him whatever and it had a Post-it note on it that said 'Don't press this button'. And like, if that Post-it note falls off, that's crazy, and why can't the machine do what it's supposed to do without, you know, the people having to write arrows and instructions on Post-it notes. So with the grant I got I bought one of these infusion pumps and I thought it was mad. It had got all sorts of quirky problems, like it had timeouts. If you started to enter a number, so I want to give you 25 millilitres per hour, so you start typing 25, let's say you type '2' and then you think, do I want to give you 20 or 25, if you paused for like five seconds, it would reset. So if you press, okay, I'm going to give you 25. So you pressed the '5', but because it's timed out, it'll give you 5 millilitres per second, not 25. And it just sort of loses the '2' because it's got fed up waiting for you. And I thought this was bonkers. And in fact I made a list of about thirty problems I found with it. I talked to the manufacturers and the best thing I did is I talked to some anaesthetists and they got excited, because anaesthetists love technology, and I, for a week, I scrubbed up and pretended to be – I went undercover – I pretended to be a porter, so I pushed patients into the operating room and I did exactly what I was told and so on, so it was perfectly safe, and I just sat in the operating room and watched and saw how they used all this equipment for real. Because I decided when I found this list of what I considered to be bugs in the infusion pump, well, to be humble about it, maybe I don't understand what infusion pumps do, you know. So my idea of – as I've just told you about timeouts – maybe that's essential for clinical reasons. I mean I doubt it, but maybe. So that's why I chatted up some anaesthetists to find out what actually goes on. So I sat and watched them. And in six big operations I was in, in every operation something went wrong with one of the bits of computing technology. Sometimes it was pretty straightforward, but one operation, the ventilator crashed. So the ventilator, the patient has been paralysed so they don't twitch when you cut things and so on, so you need the ventilation to breathe, and the ventilator just crashed and stopped working. And you know what PCs say, it said, 'Abort, retry, fail'. The anaesthetist sighed, got up and he rebooted the ventilator and then once you've rebooted it, the ventilator doesn't know there's a patient on it or what the total volume of their lungs is and all the rest of it, so the anaesthetist has to enter all this data, gets the patient breathing again, and he sat down and when everything had calmed down I said, 'Are you going to report this?' And the anaesthetist said, 'Well, the patient's alright'. So, that's an example of one of the problems we had. And also the cultural problem that these things happen all the time and the professionals like the anaesthetist just sort them out and the patient comes out at the other end and is alright, but let's say two things had gone wrong at the same time, the anaesthetist wouldn't have been able to cope, I guess. So it's a dangerous problem. So I got a small grant to study infusion pumps and then I'd moved to Swansea at this stage, I worked with UCL and Queen Mary College in fact, where Paul Curzon was, and we set up a large programme grant to study digital healthcare, see if we can make a difference.

[00:39:39]

Whose duty is the improvement and recording of bugs and faults? Meaning, is this the hospital's duty, is this the doctor's duty, the nurse's duty, the company, the producer of the medical device's duty to record and to work towards…

Well, if you imagine we live in the perfect world, it's the manufacturer's duty to do what's called post-market surveillance. They should be tracking how their devices are used, and ideally, you know, any infusion pump would be connected to the internet or something and it would be logging what's going on and the manufacturer should know when it malfunctions. But that doesn't happen. And some manufacturers are now logging devices so they can make the next model, but one of the problems in the medical domain is when there's a problem, people don't want to talk about it. It's not nice when things go wrong and people get hurt and somehow the media hasn't picked up. One of my analogies is like cars before the 1960s, cars used to be fast and they were unstable and so on, but the manufacturers were saying drivers have accidents. So when something goes wrong it's the driver's fault. And that's pretty much the culture we've got in healthcare, that if something goes wrong it's the nurse's or doctor's fault, because they should have been trained to use the medical equipment and if something goes wrong, well, they made a mistake. And it's very much like if you've got a driving licence you ought to be able to drive your car, and if your car rolls over, you lost control. And in 1965 Ralph Nader published *Unsafe at Any Speed*, which was a criticism of the car industry, and he showed how the manufacturers were basically suppressing this. And it's an engineering problem, it's not a usability problem like the handbrake on whatever car. He's got a case study of handbrakes failing and he discusses this. The manufacturers say the driver should know that you should, when you stop on a hill you should have the front wheels like this so that if you roll you go into the kerb. And he says, well, why don't you just make the handbrake work properly? And it's an engineering problem, it's not a training problem. So Ralph Nader transformed the car industry and their slogan – if you think of it like a slogan – it's gone from drivers have accidents, they're now saying, drivers have accidents, therefore we must design safer cars. So now when you buy a car it's got airbags, crumple zones, seatbelts and so on, ABS brakes. Those are all engineering solutions to help drivers be safer. And if you look at something like a modern infusion pump, they don't have things like airbags and seatbelts on them, that if the nurse types like this, mad things can happen. One of the simplest things I put on an infusion pump that would make it a lot safer is you're connected to an infusion pump, you're on, let's say morphine, and if you don't have enough morphine it doesn't kill your pain and if you have too much morphine it kills you, so you have to get the right dose. And I update your morphine, give you a new infusion. If I make a typo, make a mistake, you will get too little or too much. Infusion pumps, every time they're reprogrammed like that could say that 10 mls per hour, you had kept the patient alive until this moment, you're now giving them 100 mls per hour, is that what you really mean? So the infusion pump could spot the sudden change and then just ask the nurse, are you sure you want to change it, making it ten times larger or ten times smaller or whatever, and the nurse might say, well yes, that's the right thing to do clinically, or they might say, shit, you know, I've made a typing mistake, I've missed out the decimal point, or something. But at the moment, infusion pumps, if the nurse types 100, they say, well, 100 it is.

Do you think this is just an HCI problem and that can be solved by doing this, or that has to be approached more broadly?

There are lots of bigger issues. One is, as you hinted with my going undercover in the operating room, is doing research in healthcare's really tricky. You've got to get ethical clearance and permissions and things, and if a patient's sort of involved in your research, you've got to have their permissions and that's complicated. So lots of people stay out of research in healthcare in hospitals because of the ethical hurdles. And the other end of it is people think computers are wonderful, so why do you need to do research. And the solution to healthcare's problems is new computers, whereas if you ask me, the solution to healthcare's problem is better computers and it'll take a bit of work to find out better computers, but who wants to do that work when everybody else just wants to buy new things. So there's a big cultural thing to realise this is a problem and it's a soluble problem, but the solution isn't just new exciting stuff, it's thinking about it and thinking about what you really want to do. So one of the common problems healthcare's got is healthcare's really complicated so let's computerise it. And you discover you've computerised a mess, so you're just making the mess go faster. So you need to go in there with some plan to improve what's going on, and then you've got to convince the doctors and nurses that this improved thing is better. And then it's worth computerising it. But it's a mess and lots of people have built computer systems for hospitals and they don't talk to each other, it's called interoperability problems, it's a lot of work to sort it out and lots of people are in denial about the engineering problems there because they see it as a training problem and a, you know, we haven't… our computers are all old, we need to get up to date ones. Well, what have you learnt about making computers safer that makes it worth getting an up to date one. And the answer is, nobody knows. Well, in that case, wait until we've done the research. But that needs doing first.

[00:46:29]

What do you think are the potentialities and risks of AI in healthcare?

Well, a potentiality is it's got people to recognise our biases a lot better, which is interesting. So, you've heard that AI, I don't know, treats black people worse than white people in hospitals and it's biased racially, and there are all sorts of other biases as well – men and women and so on. Well, we've noticed the biases, so that's good. And of course these AI systems learnt from data that was about what humans were doing. The biases were there already, but the AI has made it visible. So that's a big improvement. And we now realise it's a problem and definitely it is, we need to solve it and the AI's making that problem rigorous so that it can be solved, so that's good. But, there are problems like in diagnosis and so on that, recognising breast cancer from X-rays, and there's a whole range of interesting things where some clinicians feel, well, it's competing with their authority. And sometimes the AI says something and people say, well, you just take it for granted the AI's right, but it may not be. It changes things. It also changes how we blame. So we're used to humans making mistakes, so it's no surprise when doctors misdiagnose things, but when a piece of AI misdiagnoses something, it raises some really interesting problems, like, should the clinician have made the right diagnosis and ignored the AI, then yeah, what's the point of having the AI if it can't do the right things and it's undermining the clinician. And who's liable when something goes wrong. And the big difference there is of course the doctor who diagnoses, you know, he's just one individual, whereas the company that made this thing has got thousands of employees, it's a big thing, and it's got thousands of lawyers, it's going to make sure it doesn't get sued. So there are interesting cultural and legal issues that come up when things go wrong, particularly.

[00:48:49]

If you look at your career, what are the proudest achievements of your career?

I think the most useful thing I did is I got into expert witnessing. So, I go into a court and I help the court understand the computer system and that's a perfect job for a professor, because you've got a real world problem – people might go to prison or get fined or something and you've got to get to the bottom of this problem that's really important – and in this case with computers I'm using everything I know about computers to see what has the court missed, what have they misunderstood, and it's like examining a PhD thesis, except it's about this thick, there's a huge amount of data and logs and things, and the expert witnessing I've done, I've kept people out of prison, because they ended up going to court over some issue caused by a computer, but the hospital misunderstood it and blamed nurse or nurses, depending on which court case we're talking about, and those nurses the court would have sent to prison because the evidence appears to say the nurse made a mistake and it's their fault, you know, professional negligence, and in court I've been able to say, well actually, the computer says this and the computer was mismanaged or it lost data or whatever, and you've misunderstood what's going on. So that's one of the proudest things. I've also helped some people get visas and things, like when we left Afghanistan, that was a big mess and a lawyer rings up and says can I look at some evidence about somebody who was working for the British Army in Kabul, their mobile phone says this, did it really say that or were they there or not, and the Foreign Office said they never had a visa and they had emails and so on, so it's a fascinating sort of whodunnit trail through the data to track down what's going on. So, through this expert witnessing you get in touch with lawyers and people and you ask them about the law and there's a very interesting law in Britain called the presumption of computer evidence being correct. So, exploring computer evidence in a court is pretty much a waste of time, because nobody understands computers, so there's a presumption in law that computer evidence is correct. It's called a presumption because it's not saying the computer evidence is correct, it's saying we'll presume it's correct because if you question it, well, you could question anything. If you get a speed ticket, and you start saying did the speed camera work, you know, you end up going down a bottomless pit that's actually got nothing to do with it. You know, of course the speed camera worked. So that's why the presumption in law is there, but people get prosecuted over things that are actually caused by computer errors and then getting to the bottom of those computer errors is really, really, really difficult, because under the presumption the prosecution doesn't have to reveal the information about the computers and the phrases in law, you know, if you ask for this and you ask for that, you're going on a fishing expedition, you don't actually know, so we're not going to provide it. So over 700 people got prosecuted for basically embezzling money from the Post Office, called the Horizon scandal – Horizon was the computer system – and it had bugs and it reported loss of money, which hadn't actually happened. And something like one postmaster a week got prosecuted and most of them ended up in prison, obviously they lost their jobs, but they lost their reputation and some of them got divorced, some of them had to sell their houses. It was an absolute disaster. And it's the biggest miscarriage of justice in Britain ever. And because I've been working with legal people about this presumption and, you know, it's completely out of touch with the reality of how computers work, I've done some work with the barristers working with the Horizon postmasters and that has resulted in some papers we've written together that have gone to the Department of Justice to try and fix the presumption of computer evidence being correct, because it's clearly inappropriate in the modern world.

[00:53:30]

There was also something personal you experienced in 2014 that is related to healthcare.

Yeah, 2014 the hospital rang me up and said my dad was not well and near to death, so I went to the hospital as quickly as I could, and by the time I got there he was dead, and in the morgue, in fact. And I saw him, came out, and the anaesthetist was sitting in the room outside the morgue and said, sit down, I want to talk to you. And then the anaesthetist burst into tears because he had made a mistake. And I spent quite a long time, because I work in human error and work in medical devices, so I completely understand how this mistake happened. And I asked for some information and that seemed fine. And then this information, I never got some of the details I wanted, and then I made a complaint and you then go down a rabbit hole. I remember one letter I got from the hospital that had got fourteen numbered paragraphs, and they were treating these things at quite the wrong level of abstraction, they were sort of saying, well, this isn't a matter because… and one of the details was dad didn't have a bed at some point, so one of the paragraphs was, a nurse is being retrained about getting beds for patients. Like, nurses don't need to know about, they don't need retraining to get beds for patients, you know, that's, this is solving the wrong problems. And then I got hold of the Datix report. The Datix report is the official thing, when something happens it's Datixed and people write up who the patient is and what was wrong and what they did. And the Datix report on my dad's death was all lies. And that is when I decided to do something about it and I escalated. But it was getting complicated because the response from the hospital was legalistic and the clinicians who are the people who need to, you know, improve processes were completely disconnected. That's when my wife said, let's make a digital story. My wife makes digital stories and this is a fantastic thing, which I'll tell you about with my dad's digital story. So, you can imagine, talking to lawyers you end up with a really long complicated thing and you, every sentence is justified and it gets logged, and it gets boring. You know, if I told you how my dad died and what happened and – they lost his teeth, for instance – gradually your eyes would glaze over and you'd stop listening. I mean it's all important but it's just lots of detail and what are you really trying to say. So Prue, my wife's technique is everybody's got two minutes, how are you going to get it down to a two-minute story. So, it takes, well, it took me about a week to do this, to get it down to a two-minute story. So it's an oral two-minute story and we dug out pictures of my dad, you know, being a grandparent and, you know, having a life. So I made a, if you imagine a PowerPoint thing with pictures, the pictures were of dad being a human, alive and, you know, father and everything else, and my voiceover was the two-minute key things, which is you made a mistake that anybody could have made and it was reported in a way to conceal the mistake and if you want to improve processes you've got to sort that sort of thing out. And I talked about things that could be done. And I finished up with a couple of textual words on the screen saying, you know, we want to make healthcare safer. That's the starting point, not covering up problems. And that little two-minute video is now used in training at the hospital when new people start, about how to use Datix and so on. But the key step in this process was having made this two-minute video – this is in the days of DVDs, if you remember DVDs – I posted it to the Chief Executive and said, 'I'm sure you've got two minutes'. A week later, I had a handwritten letter from the Chief Executive who said he'd shown it to the board, everybody had cried, and here's a list of things we're going to do. So, the digital story idea had transformed the defensiveness which wasn't listening to actually what do you really want to say, what's the key message, and get that into sort of ten minutes and make it personal. And it's a first person voice, it's not an abstract discussion about a clinical incident, it's about a human and it's about me, and I made a two-minute video about it and people listened, and things changed.

[00:58:43]

What happens every time your wife records a digital story? So what's the process? How are these videos used? You made the example of your video.

Uh-huh. So, what happens is, typically somebody's referred to my wife, the husband's died or something and they've been complaining, made a serious complaint about it, can you go and see them. So she will spend the afternoon with them having a chat, just like we're having, with a microphone recording them. And she'll ask questions and explore things and get them to, a bit like a counselling session, get them to be clear about what they're trying to say and they'll keep on saying things so you end up with a few hours of speech recording. And then she edits it very carefully, like, you know, there are ums and ahs and you can get rid of them and you can tighten it up, and there's digressions about, oh, the cat's barking or – well, obviously the dog's barking – but you get rid of these, like that mistake, you just delete it and nobody knows you made a mistake. So she gets it down and then works with the relative or patient, you know, whoever has got the issues, to get this to be what they really want to say. They then find illustrations of some sort and you put it together and then you say, this is yours, do what you like with it.

Ah, so I thought this was something that is used by NHS to improve the system…

Yes.

... but also it's a way to cope.

It's a gift to the patient or relative, they would then say to the hospital, listen to this, if that's what they want to do. But sometimes the process is completely therapeutic and, you know, Prue's had people call her and say this is the first time I've slept for five years, you know, it's taken the burden off them.

Sounds very interesting.

So it's very powerful.

[01:00:44]

Yeah. What do you think are the biggest challenges and opportunities for HCI in healthcare in the next ten years?

Actually, I think most people would say things like implants and AI, prosthetics, those are some of the exciting things. I think the real things are – healthcare's terribly complicated, lots goes on. I'm just an ordinary person but I've got two chronic diseases and you end up with different pills and different side effects and you end up talking to different specialists because they're two different diseases, it's complicated. And over centuries clinicians have been working in their own little worlds, and suddenly computers have brought everything together and you discover the sorts of records, say, physiotherapist keeps records and a nurse keeps record, a surgeon keeps records, you know, everybody keeps records and nothing works together and the computers just make this very obvious. And they also make obvious that each hospital is completely different from the next hospital, they're all little fiefdoms and so on and the way cardiology's done in London's completely different from the way cardiology's done in Liverpool and yet the computers join them up and the patient moves from one place to another and the patient records follow them. Sorting all that out isn't just getting computer patient records to work, it's changing the culture in the hospitals to rationalise it and re-engineer it and stuff like that, that's going to be really difficult, and it's going to be especially difficult because most people think computers just work and computers just solve problems. They don't realise these problems are complicated and understanding what the problem really is. If you just computerise the NHS or healthcare, you end up with the mess going faster. What you want to do is computerise it so that you make it better, and that's going to be really difficult, especially when people, you know, if somebody came into my lectures and said, actually, if you did it like this it would be better, I would say, well stuff it, I've spent twenty years learning how to lecture and I like lecturing like this. And clinicians are no different, we're all proud about all the work we put in to becoming experts and then people who study how we work, and how we work with other colleagues is becoming important because of communication through computers, those are huge cultural problems that are going to have to be sorted out. So you asked what's going to happen in healthcare in the next ten years, I think there's going to be earthquakes in culture and that's going to be the interesting area, not the technologies. Technologies are going to expose all these problems. Like I said about AI, AI has made the biases obviously a problem, but the biases were there already, but we just didn't know, and computers are doing the same all over healthcare, the interoperability problems. And obviously you can have crowd computer systems that are not interoperable, but most of the interoperable problems are exposing problems that were there in healthcare before computers even got in.

[01:04:11]

When we think about medical devices and they are in fact computers, there are some issues, of course dependability, but other issues can be the openness. I'm thinking about open hardware and open software and I'm also thinking about cybersecurity. Do you think, for instance, that openness should be an option or should it be a requirement whenever a medical device, for instance it's implanted, like I'm thinking about pacemakers or bionic eyes, so it can have consequences when it's implanted and it's not just something you can easily change or take out without affecting the patient. So what are your views on this?

I think that's a really good question. But there's a whole range of answers and nobody really knows what the best answers are. Take, rather than do medical systems, let's take voting systems. Usually they're proprietary systems and there's a lot of suspicion the proprietary organisations deliberately or through incompetence the voting systems, I don't know, get the Republicans in or something and the Republicans sponsor the companies, and there's whole conspiracy theories. Making that open source could solve all those problems, but there's something interesting that voting systems are used once every five years and they have to work perfectly. That means they're very difficult to test and debug and for four and a half years out of those five years, nobody's interested in them. So open source relies on a community of people being motivated and skilled and looking at it and going over the codes and saying, ah, this is where it says Donald Trump must be elected in the code and sort of… But for most of the time nobody's interested in voting systems, so a closed system thing, the company is making money selling the proprietary code and they've got fulltime engineers and so long as you trust them, then that's probably the right business model for producing voting systems and open source solves a problem that, well, it doesn't quite solve it, it just shifts the goalposts. So, very quickly there isn't a good answer for open source, it doesn't solve all of the problems of voting systems, it just changes them. And of course, if you've got open source, then the bastards who are hackers and cybersecurity attacks and so on, they're going to read the open source and say, oh, there's a weakness here. And rather than fixing it, they're going to wait until the vote's happened and then they're going to tamper with the results as a cyber attack. So open source lets people find problems, but it also lets the bastards find problems and not fix them but exploit them. So it's not obvious. One of the things I would do is some of the problems there, like fraud on behalf of the manufacturers, or the manufacturers going bust and then producing a system that's not going to work, maybe there's an intermediate. Like, rather than having open source, you put the code in escrow, which means basically handing it to a court and saying this is private, it's ours, but if we go bust then the escrow terms say it can become open and given to somebody else. And you could say, well, there's a risk any company will go bust or burn down or whatever, and we can't rely on you providing support when you've got those sorts of problems, Covid supply chains have gone or something, you must put all the code and the hardware designs that would have been open source, put them in escrow, so you can treat it as completely proprietary, and then when you get taken over by somebody else and you can't keep on manufacturing this, the law says that escrow open source specification can be given to another party, maybe for a fee they buy it or something. But it's a viable product if somebody runs it, it doesn't have to be the original people. But at the moment, of course, that doesn't happen.

What advice would you give to someone willing to pursue your career today?

Well, I've already done my career, so do something else. No, seriously, I, in hindsight I think I got really interested in solving problems and I got motivated by big problems and all sorts of things. What I didn't do, which some people are very good at, is I didn't build a community. So some people build up networks. I just sort of said, ah, this is really interesting and, so I've, you know, worked with teams of people and I got grants working with teams of people, but the grants you get, they have to be refereed by people. Where's the network of referees who believe this is a good idea. And some people who I've met have enormous networks of people. My sad story is I applied for a Fellowship, and the way the world works is these Fellowships you get sort of triaged, and eventually there are three people who get interviewed to get the Fellowship, and I was competing with somebody in photonics. Turns out, photonics has got this fantastic community of people who all do photonics and the people in the community all agree about what's important. So by the time you apply for a Fellowship, you're being refereed by people in the community who you know and they know you and so on. Whereas when I got into digital healthcare and tried to get a Fellowship, there is no community, and some people say well, is this the right way of doing it? And there's a sort of whole discussion that happens after I submitted my proposal, whereas in the photonics community all those discussions happened before the proposal was written. So my advice would be, if you get interested in something, great, but make sure other people are interested in it and you work with them and understand how they work and be in a community, build allies so that when you get your papers or your research proposals or travel grants, whatever it is, refereed, you know how the community's thinking and the community, you've worked hand in hand with the community and you're all trying to do the same sort of thing, and then you'll get support. Whereas, I think most of us, me included, when you write up a paper or a proposal, the referees you get is just chance. Sometimes you succeed and sometimes you go down in flames because they don't understand what you're trying to do. So build a community, network, go to seminars in other universities. You know, just get around the world and build a community of people who are aligned with what you're trying to do, or you align with what they're trying to do and you can help other people, you don't have to work on your own.

Thank you. It's been really a pleasure talking to you today.

[end of recording]