Oral History of Charles Davies

Interviewed by:
Marc Weber

Recorded: January 13, 2009
London, United Kingdom

CHM Reference number: X5303.2009

Marc Weber: Just begin by giving a little bit of your early background. I read an article that you're from Wales.

Charles Davies: That's right. I was—at school, I was good at physics and really liked physics so I came to London and I've never gone back. I did a degree in physics and then a PhD in computational plasma physics and that's where I got introduced seriously to computers. At that time, I was introduced to FORTRAN [IBM's Mathematical Formula Translating System] as the scientific language was then and probably still is, as far as I know. I wrote programs, simulations and simulated in three dimensions, magnetohydrodynamics for fusion research. When I did that my supervisor was David Potter, who founded Psion. As a whole, I stayed nine years as a PhD and post doc, which is salaried but temporary; a temporary lecturer it might be called in America. I left that post to join David Potter shortly after he started Psion.

Weber: Did you ever think you would end up in computers and in business when you started?

Davies: Well, computers were a very new thing and—well, not that new, I guess. I suppose the thing that was new and which was exciting and was started was really microprocessors. This was in 1981 and many computers were starting. It was the time of Digital [Equipment Corporation] and mainframes being challenged by minicomputers. It was before IBM's PC, of course, but there was this thing about microprocessors which looked really interesting. The computers that were around then were the PET [Commodore Personal Electronic Transactor], and very early home computers. It was an interest in those early home computers and really a cottage industry. No, I didn't really—I never planned my career. I never thought I'd end up particularly in computers. Also, in the UK, there wasn't the entrepreneurial spirit. In fact, David Potter did a sabbatical in the 'States where he probably got the idea, where it was very prominent on the West Coast to come out of academia and start up a company. I wonder if he'd done that if he hadn't been in the 'States for a year. It seemed an exciting thing to do and I didn't have any kids to feed. I got married at the same time, actually, in the same month that I started, September '81, and so it seemed an interesting thing to do at the time.

Weber: How did your wife feel that you were throwing away a good career in physics?

Davies: Well, it wasn't a really good career and I had a feeling that—I was feeling ready to get out of academia and that was probably a good call. It was definitely the right choice. I'm a sort of capitalist at heart and I think I'd rather be judged by the masses of people paying good money for a product rather than by referees on a research paper. Unless you were an Einstein, most people would get on well in either research or commerce with the same goal-oriented… understand what the objectives are and try and meet what people want. Anyhow, it felt right at the time and, no, there wasn't any feeling of high risk. I wasn't worried about risk and had no idea about financial security or anything. I went along for the excitement of it and the interest. Yes. I wasn't worldly wise enough to know whether it would be a good thing in financial terms or anything like that.

Weber: Had you been interested in building things?

Davies: I really liked software, certainly something happened on software. I'm not a “make things out of physical bits” [type], but even in my school—an odd school, nowhere, out in the provinces in Wales—I learned FORTRAN, actually, very early in this time, even in school. I did a course of six lessons where you filled in a coding form and got punch cards and I just took to programming like a duck to water and really liked that.

Weber: How old were you when you were first exposed to computers?

Davies: I must have been 16 at the time. That was very young because computers weren't in the public consciousness at all. It was a very minority thing that people hadn't heard of. I suppose it's reverted to that, in a sense, because almost nobody programs these days. My kids don't. My kids can't program. I was unusual and I liked the idea of computers so I was just drawn towards it.

Weber: This was something at school that you—a special project?

Davies: Yes, it was [the] availability of a course and it was an unusual thing. I found it very easy to pick up. It's true that I also picked up books on [how to] teach yourself programming. I don't know when I'd spotted computers and I thought, “This looks interesting.” I had a grasp. Things like computer variables and the difference between a computer variable and a mathematics variable I had that licked at 16, which was unusual back then.

Weber: Yes. Were you interested in music?

Davies: No. No, I liked to listen to music but I'm tone deaf and I'm Welsh and I can't sing and that's kind of unusual, too. You're meant to be able to sing if you're Welsh. I'm one of the few people out of Wales who just can't sing to save their lives.

Weber: I suppose that's why you moved here. <*laughter*>

Davies: Maybe.

Weber: And math you liked?

Davies: Maths I liked and, yes, I took to maths easily and I took to physics. I really did them because I found them easy and it's self-reinforcing. I still like physics. I still try to read a bit of physics and keep up. When my kids grew up, I had to help them a bit with the physics and I found it enjoyable to have to re-learn. I can still do partial differential equations if I have to. I surprise my kids sometimes that I can remember this stuff. I surprise myself that I can remember this stuff. Yes. I liked that.

Weber: Okay. Then you pursued physics and then there were computers around.

Davies: Yes. You use computers in lab and I would gravitate to those projects that required a bit of programming. It was sort of accidental. I went and did a PhD and I didn't think I'd picked it because it was computational element but it ended up being a computational element. I thought, “That's nice, I like that.” I picked someone who'd written a book on computational physics. My supervisor had written a book on computational physics and finite difference methods. I ended up getting heavily into programming and discovered there was such a thing as—in my undergraduate, it was still punch cards. I can remember queuing and putting punch cards into a machine. The fact that you could queue and instantly—there was something called instant turnaround where you could put punch cards into the machine and not send them off to an operator to do it. As an undergraduate loading the punch cards yourself into the machine was seen as—it was called instant turnaround.

Weber: Almost interactive?

Davies: Almost interactive, yes. We used to play a game where I had a lab partner and someone would be in the queue and someone else, when inevitably the program failed, could parlor something on—you'd queue while the other person fixed the punched card that was in error. You'd try to [make] an iteration and compile within five minutes, which seemed remarkable at the time rather than waiting a day which was what I'd been used to.

Weber: Sure. What sorts of computers were these?

Davies: Well, these were CDC computers and I remember they didn't run more than one program at a time, right? When they ran your program, that's all they did and they didn't have much memory so there was no virtual memory. It was real memory and, yes, they had real tapes and stuff.

Weber: No timesharing?

Davies: No timesharing at that time. IBMs were available but, in computing, we used 60-bit CDC machines. Actually, we did have terminals when I was doing my PhD but these were line editors and so there were no screen editors. When I started Psion, I used a home PC, which was a BBC Micro, or a Sinclair ZX-81, as it was then. With a BBC Micro, I had my first experience with a screen editor, which was kind of like a word processor where you actually interact directly with something that you scroll up and down. With a timesharing machine, you could list three lines and you literally would replace something with something even in the line. I'd had interactive terminals but only in the context of line editors.

Weber: Right. It was really unexplored territory?

Davies: Yes. The idea of having a whole processor to yourself on your desk, so to speak, as in a BBC Micro back in those days or even a Sinclair ZX Spectrum, which was what Psion started on. He started on writing software for home computers.

Weber: Yes. You had told me the idea for Psion seemed to come in an Italian restaurant.

Davies: Yes, well, [when] Psion started off, the first thing I did was publish—in those days, you had these home computers which connected to a TV and you used an audio cassette recorder to store programs on and to load them on. We started off, and it was really a cottage industry, very much an asset industry and we produced software for things like the BBC Micro. The most successful computer in volume terms at that time, '81,'82, was the Sinclair ZX80 and then ZX81. We produced software for it and published software. The first two programs we published were something called Fantasy Games, which was a single-user adventure game like [a] Dungeons and Dragons-type thing, and a chess program, which someone had written. I got that—

Weber: Graphical?

Davies: Not graphical at all, no.

Weber: A board that moves? Character based?

Davies: No, actually, it was graphical. The chess program was graphical, yes, 2D obviously. Then we started writing software for it. We did some serious software so we did things like a graphics program that allowed you to do 3D graphics and hidden surface removal. We did also a flat-file database for the ZX Spectrum where you could [create] literally, a single file with fields that you could fill data in and you could have it on your desk like a Rolodex-type functionality, electronic Rolodex. We sold that program and it sold reasonably well.

Weber: Mostly for address type—

Davies: Yes, we didn't think of it just as being an address [book]. You could keep your album collection on it, something like that. Address [Book] was one of the applications and it was just a flat file and it was called View File. When we had the idea, we were having lunch in the local Italian restaurant. The idea was basically that this was quite useful functionality, but you had this thing where you'd set up the TV, you'd load up the program, which would take five minutes, literally, and you'd load up the data, which would take another five minutes, and if you kicked a wire, you went back to square one. It was quite a precarious thing on your desk but it was useful and we thought, “Wouldn't it be more useful if you had that in something of a form factor of a calculator?” That was the basic seed of the idea. We hired someone—in those days, there wasn't quite the separation between software and hardware. You had people like Atari and it was more like Apple is now was the norm. You didn't have separate software companies and separate hardware companies. We didn't have a model that restricted us to software or hardware, and we were quite up for the idea of doing hardware, and so we hired people. We hired their expertise and we then did this concept which led to the [Psion] Organizer 1 in 1984.

Weber: When you were sitting in the restaurant, you were already doing the software?

Davies: Publishing and writing software, yes.

Weber: *<inaudible>*

Davies: I was personally writing software.

Weber: [You were a] full-time software company at that point with how many people?

Davies: Yes. It was two people at the beginning, and I started writing software when there were two of us. That was successful and we started hiring more people and so we grew. At the time we had the idea, we might have been four people or something like that. Subsequently, we moved to bigger premises and were up to 20, that kind of level. The early cash flow of the company was based on—the big hit and the breakthrough was in producing software that was sold for [the] Sinclair ZX81 which had a kind of burst of success. We sold this software—

Weber: How many copies, do you remember?

Davies: Gosh, I don't know. I honestly can't remember. I remember we'd get the deliveries and I remember piling it high and selling it. I honestly can't remember the figures. I could look it up but it was a kind of mass volume to retail stuff.

Weber: High Street?

Davies: High Street, yes, High Street stores. W.H. Smith was the main retailer in this country and it was selling that software. Yes, it had gone beyond the cottage industry stage and was selling. It was good, zero precautionary costs, so to speak, and we made very good margins and that provided the capital that allowed the hardware business to start where you need, you do need cash. We were able to do that without venture capitalists being involved at that stage.

Weber: Where was your office?

Davies: It was near Baker Street. A place called <*inaudible*>. Yes, small office, kind of funky office.

Weber: What was the atmosphere like?

Davies: There weren't companies like that and it was an open plan with bench around the side. Now, the early software I developed, Flight Simulator to the ZX81 was actually developed on the ZX81 with an add-on keyboard accessory. The ZX81 had a membrane keyboard and then we gravitated to using TRS-80s—which we called Trash 80's at that time—we wrote software there. They were eight-inch floppy disks—no, they were probably five and a quarter inch and we were using MASM, Microsoft Assembler, at that time because all our programs were written in Assembler. We made enough money to buy a DEC VAX [minicomputer]. We bought a VAX-11/750 and we were running 15 people off it with a megabyte of memory, I remember, and an RA-80 and the 80 meant 80 megabyte hard disk. It seems tiny now. We had probably 15 people and we chose the wrong programming language at first; we chose Pascal. We decided to go to a higher-level language but, after a few months, we realized the errors of our ways, and switched to C. We were influenced by the Mac—Pascal was the system language at that time. Then we

used a C cross-compiler and started writing games and stuff for that. The early development for the Organiser was done on eight-inch floppy disks with a Hitachi development console being run on it and that was done in Assembler.

Weber: Hitachi?

Davies: Hitachi being the processor. It was a Hitachi processor. Gosh, I can't remember the name of the processor now but it wasn't X86, it was before X86. It was—

Weber: It was a dedicated machine?

Davies: It was a dedicated machine for developing for that processor, yes.

Weber: Oh, okay. Got it. Right.

Davies: It had eight-inch floppy disks and it had its own assembler for that. Very primitive.

Weber: The people you were hiring—were there other software companies now?

Davies: No.

Weber: So where did you find the programmers?

Davies: They were just people from university, basically, that we got in and hired and—

Weber: Did they know how to program?

Davies: No. Well, probably not, no. We had some people who had written for games consoles, the kind of things you'd see in pubs, dedicated Atari games machines like Space Invaders machines. We didn't pick people from the industry. The industry in software at that time was mainframe people so the companies in the UK would have been IBM and ICL..

Weber: Sinclair's was—

Davies: Sinclair was a startup and so these were just people out of university who—we taught Assembler and C. We didn't get experienced people as such, we just got new ware people and we taught them programming.

Weber: There were no trade shows particularly?

Davies: There were things like the Atari around at that time but they were very small scale, almost cottage industry, driven by the growth. There was a Cambridge-led thing in the UK with both Acorn and BBC; they sponsored this computer called the BBC Micro and there was Sinclair as well with ZX81. I think parents bought these things on the BASIC—people did buy them and learned to program in BASIC and so some of the people had learned to program on these machines and learned assembler. We hired people whose parents had bought them; young people who had the ZX81 and they'd got bored with BASIC and learned that you could get assemblers for these things but you'd programmed assembler into the RAM statements of the BASIC and ran programs, unbelievably primitive. The people learned and we hired them but there was no like-minded—we were isolated and weren't making out ourselves as kind of academic, but we were a bunch of academics, really, happy making stuff up. Maybe it would have been different in the 'States, I don't know.

Weber: How did you get the hardware developed?

Davies: We hired someone who became a director, Andy Clegg, who led the hardware design. For example, I remember for the hardware in the Organiser I, [it had] unbelievably small amounts of memory so it was a kind of system on a chip with 4KB of masked ROM so that's not like Flash. You didn't have Flash back in those days and not even erasable so there you made the image of the operating system and you specified the kind of file three months ahead and then these things were manufactured and you took stock of them so that was your cycle so you have to get it right. You had 2KB of RAM and the Organiser I actually ran Forth. People don't remember that language; an incredibly tight language, and very basic functionality based on View File. That was a very early product. It had a memory interface, because it had no memory, and so the idea, we produced the first memory card, I'd say, based on an erasable EPROM, as it was called then, which is memory where the chip had a glass window on it and you'd expose it to ultraviolet light to erase it. You had, to all intents and purposes, a write-once memory card which would have a capacity of 4KB and you'd keep your names and addresses on that. When it filled up, it filled up. You could either buy a UV [ultraviolet] eraser or send it in to be reformatted, in which we would recover the deleted space. When you modified something, you couldn't recover the deleted space. It was write-once. You could logically replace something or update someone's telephone number but you'd gradually consume the space and you'd send it back for erasing and we'd copy the data off, erase it and then put it back in.

Weber: In normal use, that would be something that would happen after a week, ten years?

Davies: Oh, no. It's after a year or something because people didn't have a lot of data. I think you probably had 8KB of these. I'd have to look it up. 8KB of data would last you—that's 8,000 characters on a primitive keyboard. This wasn't multimedia content.

Weber: Okay.

Davies: This is ASCII and probably, you know, probably not even 8 bits per character.

Weber: It was pages of raw text?

Davies: Yes.

Weber: Those customers probably didn't even need that erasing. Did most customers—

Davies: The Organiser I was a limited success in the consumer space. We made 30,000 and, in fact, we were saved in that a lot of the sales were actually made as an industrial computer because we sold—we sold some of them to Marks & Spencer's in those days. There weren't electronic price codes and stuff and so they were put in a system where they programmed price information and carried these things around.

Weber: You told me they had quite an elaborate system, battery operated tills?

Davies: Yes, in those days, I believe—I've heard the story that, in those days, they had battery operated tills that had [lead-]acid motorcycle batteries operating them and I think there was a rule—that they'd had a fire that was caused by an electrical fault and so there was a rule that they couldn't have electrical power going to the tills. These acid batteries were somehow safer, I guess.

Weber: <*inaudible*> started the fire.

Davies: Yes. Quite. Anyhow, there was, if you like, a maverick manager who read out consumer ads in the Sunday Times magazine that we placed and decided to use our product to do price lookup. They had a system where we would—this is way before electronic point of sales type stuff came in that people take for granted now—centrally manage the prices and give people the prices based on the unique product codes. They'd look at some kind of ticket on the product and they could bring the up today's price and that was—

Weber: Based on a bar code?

Davies: No, no. This was before bar codes. I think they just entered a number

Weber: Oh, when they did it. Right.

Davies: They would key in a number and it would come up with the price code. That was the basic application. We produced this product for the consumer market and we discovered an industrial market for it as well. Since that time, Psion was actually producing its products and we'd produce industrial versions as well as consumer versions. Psion, to this day, is still in that industrial market, which is less visible.

Weber: So each day, presumably, they would load prices to some central place—

Davies: Yes, they had industrial strength erasers which we built. It was not high volume stuff. They had an internal mail system involving deliveries to all the stores every day and so they would centrally manage probably off some kind of IBM computer and they would download the data to these things. We had things called gang bangers, which would blow 16 of these memory cards at a time and someone manually did those and distributed them. Every day, the local store would receive the latest memory cards and they'd load them onto the computers and give them to the staff *<inaudible>.*

Weber: And send back another one.

Davies: And send back another one and recycle. We did systems programs. We implemented the system in assembly language. We knew nothing about systems analysis or any of that stuff and we just did this and this was good business. It enabled us to sell all the Organiser I's and then we got onto the Organiser II.

Weber: How many per store?

Davies: Oh, gosh, I can't remember but I guess ten or something per store. I think one per till, broadly speaking.

Weber: All of Marks & Spencer's across the UK?

Davies: Yes, I think there were 300 stores from memory. It was a long time ago.

Weber: Right.

Davies: Marks & Spencer's is still a large organization but very centralized in those days and very top down. They were very good at handling—they were very well equipped to handle the mainframe computer to disseminate this information.

Weber: I presume each Organiser represented a sort of license for *<inaudible>.*

Davies: It was a support business, yes.

Weber: Over all, this was a good part of your income.

Davies: Yes, this was very successful business and turned into what endures today. Psion still is publicly quoted and you can get a parking ticket from a Psion today. This is handheld computing and warehousing. That business grew and it's the same innards.

Weber: Does it say Psion?

Davies: On what? Yes. Psion Techlogix, if you look at the website.

Weber: On the parking ticket.

Davies: Oh, no, no, no. It wouldn't say—

Weber: <*inaudible*>

Davies: No, it doesn't say Psion on the parking ticket. These things are used [in] industrial computing. Throughout Psion's history, there has been this dual track of producing the same—based on the same architecture and operating system, hardware platforms but with very different form factors and housings because you want waterproofing and stuff like that. There are different requirements for industrial computers but the innards were the same. Had Marks & Spencer's caused this—we didn't set out saying, "There's a market opportunity, industrial." But we produced this thing for the consumer market and somebody noticed and we said, "Hey, there's another market here."

Weber: You basically got a call from Marks & Spencer's?

Davies: That's right. From a consumer ad. It wasn't something—it was emergent strategy very much. You say, "Ah, well, there may be more like that." Then we started—the Organiser was never produced in two variants but when we went to the 16-bit platform, we actually produced, in form factor terms, very different products which were designed for either the industrial or the consumer product. The original Organiser was used as is. It was exactly the same product.

Weber: Yes. Would the revenue work—did you license a user?

Davies: No, no. There was no licensing then. We just sold hardware.

Weber: When Marks & Spencer's ordered one—

Davies: We just sold them a kit. It's rather like IBM selling hardware. There was no software licensing model.

Weber: Then they were just paying à la carte when they sent the cards in to be read then. It was just a service?

Davies: Oh, no, no. They did that themselves. We supplied them with the kit and they'd run it all themselves; they had their own IT.

Weber: You just provided—

Davies: We designed the—

Weber: <*inaudible*>

Davies: —bespoke software and we produced the system and we produced the software that ran on a desktop PC at that time. I'm not even sure it was PC but some kind of desktop. It might have been Sirius or something in those days. I'm not sure. We designed the system. We didn't operate any of it and we provided the support.

Weber: You were obviously somewhere charging much more than you would for a consumer?

Davies: I'm sure we did. We probably sold a lot and sold it in bulk. Of course, we sold it to consumers through distribution—I don't think the pricing was—for the short run and we'd charge for the software, the N erasers, the software to be written but, no, this is basically—the business model was to sell a device at a gross margin and I think the gross margins were comparable because they could go out to Dixon's and buy the same product.

Weber: Sure.

Davies: The gross margins were comparable. In fact, we'd have been selling them at wholesale prices, so to speak, because they were buying in such volume. But it was still a good market. We were making decent margins.

Weber: What did they sell for in those days?

Davies: I think it was £100 in those days, which was a lot of money.

Weber: At the time, personal computers—

Davies: Personal computer—

Weber: <*inaudible*>

Davies: Personal computers were £100, like a ZX81 was a £100 or so, which was more in those days in real terms. They weren't cheap.

Weber: You did see it as this was an alternative computer platform. Not a less-than computer device.

Davies: Yes. It was a computer in classic terms with input/output and mass storage but it was solid state and so —it was very much solid state so we were thinking—microprocessor. Rugged and no moving parts except for the keyboard. No rotating, no hard disk type thing.

Weber: You saw it as an alternative form of computer?

Davies: Yes. Computer architecture inside but an appliance. When we made the Organiser I, we thought of it as an appliance so it was something that people bought that had an internal computer architecture. We didn't think of it as providing a platform for running applications, although that did come later. The starting position was very much an appliance rather than a handheld computer. The Organiser I had no extensibility at all. It didn't even have an RS-232 [serial interface]. In the Organiser II, people wanted to connect it to things like bar codes or temperature gauges so it became more computer-like as we evolved. We never thought of people writing applications for it so we actually published—we didn't have an API in the eight-bit operating system that we used for the Organiser but we published some system interrupts so that people could add some native code and we figured this was a useful thing. It's only when the 16-bit, which was x86 based. Series 3, was based on the work-about which was the industrial equipment on the same platform. We designed it with an idea of providing a separation between the applications and the operating system. We didn't have the idea of an operating system really. We just had a microprocessor that ran software. It became more computer-like as it evolved, as we saw people could do useful stuff. It became a handheld computing platform and people did write a lot of applications.

Weber: On the first one, though, you couldn't *<inaudible>.*

Davies: You had a Forth programming language so you could write Forth applications on the Organiser I. Then, for the Organiser II, we invented our own language called OPL, Organiser Programming Language, which was actually very successful and used on the 16-bit [version]. It was BASIC-like, a bit procedural. BASIC with no line numbers. Line numbers are since going out of BASIC, I might say. But, yeah, and you could program in C.

Weber: The idea for the form factor, how did that—?

Davies: We formed an early relationship with an industrial design company that happened to be very close physically. They were called Therefore [and we started] a relationship with a particular designer called Martin Riddiford who did all our designs. Therefore is still running, a very interesting design company. We had a long-term relationship. It became Therefore, and we helped spin them out. I think it was called Frasier or something. We formed a relationship with a particularly talented industrial designer and he came up with the idea of the pull out keyboard and did our series 3s, they've got a very distinctive way of opening and a very attractive design. Sometimes there were mechanical challenges in terms of the

implementation of the tooling and everything. We had that industrial design ethic and very good long-term relationship with a particularly talented industrial designer.

Weber: Somebody had the idea you needed a designer?

Davies: Yes, it was an idea. That's true. It probably came more David]. And Andy Clegg, who came on the electronic side of the project. I was only strong on the software side.

Weber: Do you remember the first prototype?

Davies: The first prototype would have been a big board—

Weber: I should probably say, the mock-up.

Davies: Yes, I did go along and the company was small enough and <*noise*> I don't—yes, there were models and so—

Weber: Can you turn that off?

Davies: <*noise*> It will—I can turn it off. It will stop and then probably not do it for several hours.

Weber: Just wait for it to—so you were saying you did go along to the meetings.

Davies: Oh, yes, definitely I've got involved in the industrial design and that was very much fun and Martin would build models. There was no CAD in those days, and he still never liked the CAD stuff. The models were built and he'd have a variety of concepts and we'd look. That was always an exciting part of the job.

Weber: He would build models out of what?

Davies: Well, bits of plastic and glue and balsa wood and paper and cardboard and you kind of hold them and try them out and tried out sliding mechanisms. There'd be something that illustrated the mechanism based out of<*inaudible*> It was very much model-oriented.

Weber: Nobody has any of any of them anymore suppose, the original models?

Davies: No, well, I don't know. He might. You might want to talk to him actually.

Weber: That would really be a neat piece of—

Davies: Yes, I can put you in contact with Dan afterward.

Weber: There were no big alternative designs the—basically the form factors were the first representatives—?

Davies: Yes, I don't remember the competing alternatives. The concept was developed and the actual sliding out and the idea of protecting it in your bag. Those were the discussions. Yes, you really should—that's a good point. You should talk to him if you can get some time with Martin, he probably can remember. I'm sure he can remember. There's a lot of history.

Weber: The Organiser I was out for how long?

Davies: The Organiser was out at the end of '84 and I think it would have been a year later. No, maybe two years later, '86, that the Organiser II came out. We went through several models of the Organiser II. It came out in two lines. The Organiser I had one line and lot of scrolling—horizontal scrolling and the Organiser II had the top slot. Which you can see there, which is a sliding connector—and four peripherals and it came out increasingly more memory, both ROM and RAM, more applications built-in and more lights culminating in the fore light and the applications that were added when you have more lines where you managed to do a calendar application and calculator. The calculator was always in the Organiser I, it always had a calculator. You could buy other applications to run on it although there wasn't a huge market.

Weber: You wrote them all in-house, was <*inaudible*> open?

Davies: I think there was. People used to add software because we didn't publish the API—well, there was no API but we found a need to publish the interfaces—the software interrupts which call the operating system services. People did add software but it wasn't a mass market. It wasn't an aftermarket thing. It was people doing probably industrial type applications and connecting them to theodolites and stuff like that.

Weber: Okay. It was mostly in-house. The multi-lined screen—the calendar <*inaudible*>.

Davies: Yes, I don't actually remember—we did a spreadsheet on it. Yes, that was amazing. It's got a four line display; we did a spreadsheet. They said it couldn't be done but we did it and I remember that. I think that was an add-on application and people put things like The Bible on it or dictionaries. There were add-on applications.

Weber: You add add-on by putting <*inaudible*>

Davies: No, the memory cards. It was programmable with this OPL so people could write their own programs and put it on. You could either program in assembler or 6301 was the processor, which was an 8-bit processor.

Weber: *<inaudible>.*

Davies: x86 was a 16 bit-operating system. The Organiser was 8-bit. The Series 3 and the Workabout were 16-bit and that was a proper OS—a multitasking OS which is an interesting story in itself. At that time the only operating system was [Microsoft] DOS and people couldn't believe we'd do something other than DOS. We wanted a multitasking operating system, so the feature of the Series 3 was that you could switch between tasks and it was generally multitasking. These were in the days where PCs—you'd run Lotus I, II, III or whatever and when you want to run something else you'd exit that and run something else. There were these terminate and stay resident programs if people are old enough to remember those, and that introduced the idea that you can run something without exiting the main thing that you were on. This is way ahead—we had multitasking where we just ran something. We figured that computers were basically imprinting, that people who had these appliances had higher standards. This idea that you exited something before bringing something else was just crazy, so we thought we had to have multitasking so we did it.

Weber: How did you start development?

Davies: This would have been see, late '80s that we started developing the EPOC16 [operating system].

Weber: Because the Organiser II was from—

Davies: '86 to the late '80s and there was no overlap between the Organiser because the Organiser carried on selling for a very long time in industrial applications because it was a field replaceable unit. If it broke, you threw it away and got another one because they were sufficiently cheap and people have written systems to work with it. Long after it was dead, so to speak, as a consumer product, it carried on as an industrial product for a very long time.

Weber: What would a typical use be?

Davies: Parking tickets, data logging in the field, warehousing type applications, barcode attachments.

Weber: You had a barcode reader.

Davies: Yes, that would work in top slot and we wrote our own barcode recognition algorithms and stuff. We did everything.

Weber: So how big was your software team?

Davies: Well, it's scaled—

Weber: Let's take a snapshot mid '80s.

Davies: At the time Symbian started, which was '97, a hundred people went over to Symbian so I was thinking it was way less than 100. It wasn't big. It was—

Weber: <*inaudible*>.

Davies: Doing things like [a] 16-bit operating system [required] only 20 people. Small numbers in those days but working very hard. Not massive teams.

Weber: Oh, it's incredible.

Davies: Well, of course, the amount of software, I mean, you look at a software on a mobile phone these days and it's 100 MB, very high-end smart phones. We had ROMS that were 128KB—what started the first run of the Organiser I was 4KB ROM and it it did track Moore's Law. Moore's Law says it grows by a magnitude every five years if you work it out and so the amount of software these days is huge. On the other hand it's more expensive to write little—the capacity of what we have went up in powers of two and so in shipping and product we'd go to enormous lengths to fit into whatever ROM we had.

Weber: The amount of time per line of code—

Davies: Was much greater and an awful lot of effort went into trying to do things in less code and so it was a very special kind of programming. In the early days of MOS ROMs we eventually got to Flash. But that was a long time. It was pretty scary stuff because you had to get it right.

Weber: Did you ever have something come back <*inaudible*>?

Davies: Oh, sure. We'd have to do patches and we had a way of doing patches and with the Organiser we even sold them applications—we gave away an application which when you put it in, loaded a patch which fixed something in the ROM. We had some tricky times with the very harsh environment that we worked in. This thing was in MOS ROM, an unbelievably severe discipline.

Weber: Your team became the specialist.

Davies: Yes, and you can't have brown trousers <*laughs*> so you really did really have to have the quality ethic because there were severe financial penalties in getting it wrong.

Weber: Did it get competitive about who can squeeze the most out of—

Davies: Oh, yes, and you got kind of good at writing—you look at the assembler that the C produced and we'd hunt around for C compilers that were good. All the Organiser I stuff is [in] assembler. We all had an assembly language mindset and would look at the assembler and look at the code that was produced and do data compression. I remember sometimes I'd go on holiday and come back and find that we were over the ROM budget and that was bad news. We'd have to pull something out of a hat to get under the ROM budget and we found—we put something like half <*inaudible*> coding in at the last minute to try to retrieve some data. We had things like resource files and we managed to both compress the data and make it faster because even though it took more computational effort to decompress the code, the I/O time was reduced because you were loading less data. It was systems programming where we'd look—we programmed our own floating point functions. We did sine and cosine. There's no library for that kind of stuff in those days, so we did all of that.

Weber: Because a number of you had academic backgrounds—

Davies: Well, we had the maths and we had the idea that there was some literature somewhere that would tell you how to do it. I can even remember the book Cody and Waite, Chebyshev polynomials is the way you do cosine. Not many people know that because not many people have to sit down and wonder where they get a cosine algorithm from. It's there in the library, so this is bare knuckle programming, in those days.

Weber: Was there a, kind of, *esprit de corps*?

Davies: Oh, yes—we were crazy and I worked all-nighters and we would all-nighters. It was crazy and deadlines, shipping deadlines. It was insanely tough.

Weber: You have a family, so—

Davies: Yes, my wife didn't see me after the marriage for a very long time. We hardly took any holidays. An absent father type thing but my kids have forgiven me for it.

Weber: How late did you have kids?

Davies: My oldest is 23, so I had kids pretty quickly. I was doing this with a young family, working long hours. Progressively, you became less <*inaudible*> over time. It isn't productive, actually, to work all-nighters, it suggests that you—it's the kind of thing you do under deadline pressure. It's not ultimately the best way to write code.

Weber: Did the company go away or have parties or—

Davies: Oh, yes, the spirit and the passion that we had was just unbelievable. We would say things like, "We can't do this," and the team would say, "Yes, we can," so the level of energy and commitment from the start was very—and it helped that their managers were down and dirty and involved, so to speak. If they were suffering, the management was suffering, so it was a hands-on staff because we weren't from a management background. We'd come in and there was a closeness between management and it's a small company. That's really good. I'm sure it's the same in Silicon Valley startups. It was like a Silicon Valley startup but the unusual thing: there weren't any others and so it would have been easier. None of these stories are alien to anybody who worked in Silicon Valley but it was alien to London at that time.

Weber: When you went out and had dinner with friends not in that, did they—

Davies: I didn't know anybody else who did this kind of stuff. Technology startup. We had a relationship with ICL and worked with them on something called One Per Desk which never—it was an aborted project but they were very understanding. The normal computer industry in the UK at that time is so different from anything we did and so we couldn't have hired anybody from that—

Weber: *<inaudible>.*

Davies: —well, in the sense that they were doing proper computing programming and would do requirements analysis. We'd write specs and stuff but we weren't trained as software engineers. They did have terms like systems analysis. You'd have systems analysts and teams of people doing things and there are things like—Brickman had a software industry which was a service industry and had people like Logica putting in government systems and data processing systems and that was completely alien. We'd just get in there and write it and do what they would regard as incredible productivity. The nearest thing that came out of it was the games industry. We wrote game software, clue twitch games for things like the ZX81. In the UK, the games industry happened, and it's still quite—they were the same kind of caliber of people who would do amazing things in assembler. They'd take full advantage of the code rather than a software engineering or computer science perspective as they might have been called then.

Weber: Some of them were around one?

Davies: Yes, and that grew up but that was subsequent to that and you got things like Electronic Arts and the kind of big companies *<inaudible>* development. When we started, we didn't feel like there were other companies like us.

Weber: You didn't have any particular relationship with Silicon Valley.

Davies: No, no. That's true and we tried to sell in the 'States and didn't do terribly well to the 'States because America is different. From a British company like ours. We speak the same language, but the market requirements are quite different. I had interesting discussions with Donna Dubinsky at Palm in the early days and Jack Watkins and I remember fondly those meetings and they were very friendly meetings. People like GeoWorks at that time. I'd go over and talk to them and see if there was anything we could do but it never lead to anything. Certainly an awareness of all of that and a lot of our people ended up at

Apple. Some of our electronics designers designed the iPod, so people made it over there to the States but at that time, no. We weren't part of a cluster. The advantage that startups the States had is that there were other companies like them and you had the intermixing. We were on our own which was both a strength and weakness because there was no herd mentality because there was no herd. We were more innovative because of that. There's nobody to copy and no received wisdom. Innovation seemed natural to us. We were scientists in background or academics. We weren't people who'd left an industry and setup on our own. We'd started something straight from academia. We didn't know anything else than to make it up ourselves; we thought that was natural. We were innovative as a default.

Weber: So—

<audio break 10 seconds>

Davies: Not at the time because we accepted it. In retrospect I realized that we were—well, it was both a strength and weakness because if you're on your own, you tend to be more innovative and less influenced by other people. I've certainly seen that it would have been a benefit at some time to be part of Silicon Valley. Silicon Valley is being the center of IT innovation and to be closer to that would have been an advantage. There was a time when we started with the Organiser, PCs did that. The Organiser in 1984 really pre-dated the home ownership of PCs or the wider ownership of PCs and we regarded the PC as a peripheral when it emerged and we always sold PC connectivity as a peripheral thing. Even on the Series 3 we had a calendar application where we thought the competition as a paper based people—we were trying to persuade people to move from Filofaxes to our product. We never dreamed that people would be doing calendaring on a PC and people didn't because you had this—they were single-tasking. You can't do calendaring unless you have multitasking. Sidekick, I think, started with calendaring and then you got things like—I remember Threads which was a British company that was bought by Lotus and became the calendaring in Lotus Notes. Things like Outlook came quite late. We always regarded our customers as people who didn't use PCs and when PCs were used it'd be their secretaries that used the PCs. They didn't have typing skills, QWERTY wasn't something that anybody with money knew about—money to afford these products which were still quite expensive in this time.

Weber: The Organiser II sold for what?

Davies: About £100 in those times, which was a lot of money.

Weber: Same as the Organiser?

Davies: Yes, Moore's Law, basically. The price stayed the same and what you got for it increased. The price was stable. One of the disadvantages of being in the UK is that we didn't see IT trends. IT trends basically happened in the 'States first. The idea of turning around the relationship between an organizer and a PC so that the organizer or the PDA [personal digital assistant] is the peripheral, the PC is the central thing, happened in the 'States first. We missed that market trend. Being in America would have had the advantage of Silicon Valley; not only have you got other people you can network and bounce ideas off in a community and cluster effect, but you'd also immerse in a market which was more advanced in terms of adoption of IT than the UK. One of the key ideas in the Palm, for example, was it was empirical

to the PC and was designed for synchronization right from the beginning and based on the premise that you would keep your calendar on your PC and take away that calendar. I've seen the ebb and flow between—it's pretty interesting the ebb and flow between keyboards and pen inputs. It's moving now towards pen again with the iPhone and often our pointing screen. But it's moved back and forth several times in my career. One of the reasons that pen input worked is because it was supported by the ability to use a QWERTY keyboard on the PC. I remember the sales pitches—I was there—would say, "Okay, there's a thing called Graffiti that you have to learn but don't worry about that. You can input your data on a PC." Today, it's cloud computing, going where you enter it into the cloud somewhere and it was automatically synchronized wirelessly to where ever you are. That's today's standard.

Weber: You switched to the QWERTY keyboard—

Davies: We started off with an ABC keyboard on the Organiser I and that reflected really socially. People didn't have exposure to QWERTY keyboards. Then people used PCs directly, they were often typewriters and people recognized QWERTY keyboards. We recognized at the time, it was time to do a QWERTY keyboard and we went to the palmtop and a glasses case. We call it 'palmtop computing' where it opened like a small notebook and it was still small—that was really successful with Series 3 and that was based on a 16-bit multitasking operating system, not DOS, which we invented ourselves which was influenced, I think, by VMS. The innovations we did with the battery—we were really strong on power management and battery was king. Battery life was everything. We had a standard process and we actually had an operating system that stopped the clock on the processor when in idle. We had asynchronous events where if nothing was happening, the operating system knew about it and no power was drained. We were really good at that in those days, way ahead of our field.

Weber: A typical user would have to recharge how often?

Davies: Well, the batteries weren't rechargeable. They were AAs and they would last weeks on AAs and that was really important.

Weber: Because of the type of memory, if they ran out—?

Davies: That's right, there was no Flash memory in the early days but we had static memory, all right. There was DRAM, which kind of cycles to refresh, and you had static memory which is kept alive by a capacitor and lithium. We called it Larry the Lithium and Series 3 had a lithium and it kept the memory. It wouldn't have worked with DRAM but in those days you had static memory which took very little current to keep alive. Not the plug-in memory cards which were solid-state.

Weber: Users could not put data on your plug in memory cards?

Davies: Oh, yes.

Weber: Oh, they could?

Davies: They could, yes, but if you bought the product you could keep memory internally and it kept it alive with the lithium.

Weber: It would die if you put it in a drawer.

Davies: If you put it in a drawer and left it a long time it would lose the contents, absolutely.

Weber: The Series 3 operating system was how many years ago?

Davies: Well, I'd say probably two years in development for memory. We did it from the kernel, the scheduler—the multitasking scheduler, asynchronous events, object oriented in those days, very early using Influence by Objective-C, Brad Cox's book on Objective-C. Applications were processes and we had a window server. We wrote our programming and multitasking OSes from VMS and had servers for things—we had a file server. The architectural sophistication of what people regarded as appliance was actually way ahead of its time and everything that grew. We had overlapping windows when Microsoft was tiled and we had a window server and redraw events, all in 128K ROMB. It was just astonishing what we did in very little memory and gray scales. No color in those days; we had black and white and the first thing we introduced grayscales and the proportional fonts and word processing and spreadsheeting. We did a whole window system.

Weber: The first Mac was what, 128KB? But that had no multitasking.

Davies: Yes, it was software interrupts. No, this is full of multitasking. Time server internaling. We still have that today in Symbian devices. The time server programming; when something drew to a screen it sent lightning inter-process communication to a windows server which is still there in Symbian phones. On a Symbian phone today you'd probably run in excess of 100 processes—very scalable, client robust, client server which was important. One of our things that we were aiming for was we'd run a long time, the idea of "press the big red button" to reset it. You would not leave your PC on in those days. You switched it off every time. We hit the big red switch, as it was in those days, during the day and we would run for a year at a time without a reboot. We had error handling to trap memory leaks and we used to have tools that soaked up all the memory. We really did, in those days, try to write software. We sometimes would need a reset and this thing would crash but a lot of users go a long time never thinking of rebooting their machine. We just ran, we were never off. Off meant that the processor was stopped and the screen didn't show anything. The amount of current we drew was minimal because you had static memory and you almost did nothing with them.

Weber: Months.

Davies: Yes, months between system reboots. We had no memory leaks, nothing like Windows. Even these days, people reboot servers just to clear up the memory leakage. We were programming to a very high level and we didn't know anything else. We just knew that we wanted an appliance that didn't cause any problems. That was zero maintenance, I suppose you'd call it.

Weber: Why didn't you ever try to port it to a larger machine? Why keep it—?

Davies: Well, we did. We produced what's called a Netbook and that's come back, it was ahead of its time and it wasn't a commercial success but we did produce—

Weber: What year?

Davies: Late '80s, '90. Monochrome was solid state and selling for £400 in those days. I remember it was a lot of money but people bought those and did word processing on them. We even produced a DOS version. We were thinking of doing CP/M and Bill Gates made a personal sales call to persuade us to use MS-DOS.

Weber: This was before you came out with—?

Davies: This was, yes, when we developed it. We produced it and it was not a commercial success but it was an early Netbook concept and we used the term Netbook.

Weber: Just one thing I did want to ask you about <*inaudible*>?

Davies: Yeah, so it was called a Netbook but it wasn't connected wirelessly or on the Internet or had an IP—no one thought that the Internet protocol was an interesting protocol at that time.

Weber: But why did you call it Netbook? Was it on a network?

Davies: No, that's interesting question. We used the term Netbook, that's right. Why did we do that? I'm not sure. I can't remember, but we did coin the term Netbook.

Weber: And you didn't <*inaudible*>?

Davies: It had a spreadsheet, word processor, and the idea was that it was—well, in those days and still today you take for granted now that you put your notebook on standby and sometimes comes out of it and you're having to reboot. In those days the transaction time cranking up a PC was very high and even a notebook computer was very high. You'd turn it on and you'd wait several minutes. The idea here was the instant on, it was never switched off. A Netbook was like the Series 3, it was instant on. Just open it and use it.

Weber: Essentially it was a Series 3 with a larger screen?

Davies: It was the same, yeah, EPOC16 [operating system], so it was an 800 X 600 screen.

Weber: In the late ’80s?

Davies: Early ’90s, I think. I could check. No, I think you’re right. I think the Netbook came later when it was wirelessly networked in Wi-Fi and so we did produce—based on EPOC32—and that was called the Netbook. I remember now. The early Notebook form factors were called mobile computers and that wasn’t a commercial success. That was based on the same as the Series 3.

Weber: Do you have any of those?

Davies: I’ve got pictures of them in the annual reports but no, I don’t. Someone must have one. Mobile computers, yes. Architecturally, they were the same but we designed the software so it would work on a large screen.

Weber: The Series 3 was a big commercial success?

Davies: Series 3 was a big commercial success. In those days first—I guess we sold two million in the lifetime but those were good numbers for us at that time. There are three ways of operating a system or operating software. There’s the 8-bit for the Organiser, the 16-bit where we introduced object oriented in C using our own tools to preprocess. That was helpful in client server computing, multitasking, and then those ideas were carried over and that was on the x86 and then—

Weber: You mean the internal client server.

Davies: Yes. Internal client server. At that time, there was this huge debate over RISC computing. If you read *Only the Paranoid Survive*—the story of Intel—Intel had this thing about where the RISC <*inaudible*> and they were paranoid wondering whether RISC would wipe them out and in the event, they invested in RISC as well. I would think a lot internal RISC processors would be printers. It turned out to be both were important in the long term and when we were doing the 32-bit operating system—which is what runs in Symbian’s products today—we changed processor architectures and we were very much believers in the RISC thing or portable which was spot on. We did a beauty parade of all the RISC processors at that time, MIPS, PowerPC, remember that? ARM which was a spin off of ARM stands for Acorn RISC Machines and there’s spin off from Acorn that was part of the UK computer industry. We picked ARM on the basis of joules per instruction; it was the lowest power, most efficient and so watts per MIP, we decided it actually doesn’t matter. It’s the basic joules per instruction. We thought that was the most efficient and so we made a big bet on ARM and boy was that a right bet.

Weber: That’s when you had Ron or Rand [ph?]?

Davies: Rand, yes. We met with Rand because this thing called TCP/IP started to be important. This protocol just looks like a dead protocol, was something that would be wiped out by SNA, which was IBM’s system network architecture. It turned out to be an obscure protocol. People were asking for it and so Rand had done something in TCP/IP so we—

Weber: Wasn't he also involved with the processor itself?

Davies: Well, Rand's history was that he worked for Acorn and ARM, and he actually wrote an application—an alternative add-on application, I think, a calendar application for the Series 3—and then left and started up his own business in the TCP/IP that turned out to be ultimately a web browser. Actually, was it a web browser? I think it was just the TCP/IP stack.

Weber: When I saw him, he said they did have a working browser for the Psion around '93.

Davies: That would have been Psion Series 3 probably, which was selling at that time, and we were in the throes of working on the Series 5.

Weber: The ARM came after the Series 3?

Davies: No, ARM came in Series 5 and then we made a big bet. It was probably premature with the C++. The C++ was quite new then. We liked the object orientation but we wanted a language that would do object orientation. We went with C++ ahead of really where the proper support was because it was hard to get decent C++ cross-compilers at that time. Anyhow, we did do it and that was ahead of its time. Probably too ahead of its time. But we carried forward along with the client server ideas and Moore's Law was doing it's job and the amount of software was greater. At that time Flash was coming in and the idea of doing Flash. Long [–term] volatile storage really only started then. The idea of not using RAM for system storage, we started that. No one uses RAM for system storage anymore but we moved on. We got to the stage where we had about 100 people producing software for Psion devices that we decided to divisionalize. Within the organizational structure, we had Psion Computers which did the consumer products like the Series 3. We had Psion Industrial which did the industrial products for things like warehousing and getting a parking ticket. Data collection, I suppose, a lot of the applications came down to. We separated out the software part into Psion Software and charged an internal royalty because [it took] a significant amount of R&D keeping the software going. We made a big decision. I can remember the strategy meeting on whether to license the software because it's becoming expensive to fund a software platform given for our own devices with the internal royalty getting high. The management team went out and did their strategy-away day and decided that PDAs were uninteresting and not with any future in it. Mobile phones were looking very interesting so they went off to mobile phone companies, as market, and the rest is history and that's how Symbian ultimately started.

Weber: That was around mid-nineties that you first—?

Davies: No, early nineties, really. Early nineties, because Symbian was spun out in '97.

Weber: There was five years—

Davies: The selling—I can remember going to Finland and visiting Nokia and giving a presentation to stony silence. The Finns aren't always the most—they don't give you the feedback. I can remember giving them a presentation and then asking them what they thought of it, and they said "We ask the questions."

That's it. This is a time I tell them about; I wrote them about that and they loved the story. This is a time when Nokia produced its first communicator.

Weber: <*inaudible*>

Davies: Yes, that's right. The first Nokia communicator was based on GeoWorks and it was a clunky device. We saw that mobile phones were something kind of interesting and we connected our products to mobile phones. Things like the Nokia 2110.

Weber: Do you remember the period when somebody identified mobile phones as—?

Davies: Yes, we identified mobile phones as interesting things to connect to.

Weber: When, late '80s or early '90s?

Davies: It would have been the time of 2110 whenever that is. I guess that's the early '90s.

Weber: You haven't thought much about networking in any context?

Davies: Actually, we thought hugely about wireless networking.

Weber: <*inaudible*> vertical apps.

Davies: For vertical apps, yes, because there were things like Mobitext. You think of Blackberry and RIM. That started off as two-way paging. We were into paging and we did paging on the Organiser I.

Weber: Right. That's the thread I wanted. Then you started doing that way back.

Davies: Yes, we always had wireless but it was more industrial wireless and we saw the GPRS [General Packet Radio Service] coming and we saw it way before. For the amount of gestation time for the GPRS to happen and to really work was just unbelievably long. I would have just cried if I'd realize how long it would take for mobile networks to get GPRS working.

Weber: That came with GSM, right?

Davies: Yes, It was part of a standard of GSM so you could read in the manuals that there was this thing called GPRS that was going to wipe out—around in August I remember, with two operators in this country that did Mobitexts with its own kind of standard, but it was really weird stuff, a lot of stuff. People were using pagers in the 'States for a long time. I remember arguing with people saying, "How can you believe

paging is going to happen?" Of course, selling lots of Motorola pagers for an unbelievably long time, and it took forever for SMS [Short Message Service] to come in and replace the paging.

Weber: Chronologically, on the first Organiser you already had some wireless communication.

Davies: We had people come to us saying, "We can connect a pager to this," and that was interesting. We were involved with wireless networking all the time and then Wi-Fi happened but that was way later.

Weber: Right, service steps would have been what, early to mid '80s?

Davies: Yes, I'd say even in '86 someone was talking to us about connecting the Organiser to the pager.

Weber: The Organiser you think was connected to that?

Davies: Definitely connected to the paging. Why buy a sort of value-added resale but not by us?

Weber: For what sort of use, too?

Davies: Right. <*laughs*> But I remember people talking to us about it and they would have used the interrupts and stuff to do that. It's probably in the annual report somewhere. You can look. Then you know there was this idea; we realized the value of wireless because—that's right, we call it batch computing because batch computing meant that you had to do what Marks and Spencer's did which was basically data capture. You take the machines back, dock them, then process the results and of course it was batch versus real-time. To do real-time you need wireless networks and we did. We ultimately, of course, did Wi-Fi but it was leading edge. We even did data cards and there were things like GPRS data cards, but GPRS took horribly long to happen and to work. There are even places where it doesn't work. I was on holiday in Tanzania. You get sort of a cover but you try opening the Web browser and it doesn't work. It depends whether the operators put the infrastructure in.

Weber: No, there was coverage in the San Francisco Bay Area.

Davies: <*laughs*> Okay. The reason Blackberry got in with the push email was because it had Mobitext which is basically SMS on steroids. It's about a 1KB packet and I think their architect is still influenced by the Mobitext, which is an Ericsson standard.

Weber: Oh, really?

Davies: Yes.

Weber: Not by any international—?

Davies: It's designed by Ericsson.

Weber: <*inaudible*>

Davies: No, Mobitext was invented by Ericsson. I think it was an Ericsson proprietary radio standard. Those are the kind of—yes, we did that. The radio networks never took off. Then this thing called cellular happened. And the rest is history.

Weber: Because Palm also did the one <*inaudible*>.

Davies: Yes, they probably did that too, RAM and ARDIS in America. Those were the two networks. I think ARDIS was IBM and somebody else actually. We were wishing for radio coverage of wireless connectivity. Then Wi-Fi came. We acquired a company called Teklogix which had a lot of Wi-Fi skills.

Weber: That's end of the '90s.

Davies: Yes, end of '90s.

Weber: When was the first wireless, because you said third party and other people added it to your product. When did your service actually integrate your wireless standard?

Davies: Wel,l we never did and it was always—we were never a wireless company but a company that used wireless, and so we put PCMCIA slots ultimately with wireless. Into the industrial computers we integrated wireless modules into the products but we never—there are very special skills for doing wireless.

Weber: <*inaudible*>

Davies: We put Bluetooth into the Series 3; it was the first actually wireless-integrated product.

Weber: More for controlling local devices though? Typical Bluetooth.

Davies: Yes, talking to local companies.

Weber: Right.

Davies: Big applications.

Weber: <*inaudible*>

Davies: Well, to use it as a connection. The wireless we did—for example, we were very early for the 2110. For Series 3, we did a cable that connected to a mobile phone. One of the applications was that you had a QWERTY keyboard to do your SMS and do email. We did email, POP email, on Series 3.

Weber: Using mobile phone <*inaudible*>.

Davies: Yes. We also did sell modems.

Weber: Wired modems.

Davies: Yes, and do email.

Weber: I mean which Series?

Davies: Series 3. Certainly not the Organisers, but the Series 3 did email, public email.

Weber: Okay, so early '90s.

Davies: Yes, and we did that so you could do that with a phone and we just saw that as RS-232, practically. Serial connection <*inaudible*> type stuff, yes, we did that. We had an email application. But we never integrated it. We did look at that and we looked at buying a mobile phone company. The big argument, strategically, that we had at Psion was would PDAs be wiped out by phones. We did see that coming and ultimately they did. The way we tried to handle it strategically was a strategic alliance with Motorola which fell through. Which went all the way through, likely when they got there. Then they canceled the project.

Weber: When was that roughly?

Davies: That would have been about 1996.

Weber: But you were already talking about the—

Davies: Three years before that starting. Trying to sell. Talking to Nokia, we licensed Nokia's interface to be able to talk to their phones. I know people in Nokia now who can remember licensing us the interface

to their 2110 which we had to pay them a lot of royalty for. We were selling the proof to talk to their device and we had to license the interface specification. Those were the days.

Weber: Charging to use their API?

Davies: Not even their API, it was just they wouldn't give us a document that said, "Put the pins in it," and what the protocol was.

Weber: Just for the physical connection.

Davies: The physical connector and the protocol for talking to that connector, yes. We paid them money for that. I've written them about that, too.

Weber: Now that would be difficult to account for now that you're part of that —

Davies: <*laughs*> Right, but in those days interfaces were proprietary. IBM would do the same. You plug compatible. We could have tried to reverse engineer it, but we chose to license it.

Weber: You did it with them but why not Ericsson or Motorola, or did you try?

Davies: Good question. Ericsson was as big and we probably did try with Ericsson.

Weber: They were bigger at the time, weren't they?

Davies: Yes, so I would say that my recollection at that time that Nokia, Ericsson and Motorola were comparable. Ericsson might have been the biggest of the three. How things change.

Weber: It was more you were just really looking for a phone company to team up with and just—

Davies: Yes, we saw phones as interesting. In our strategy meetings we were looking at how we could connect to phones. We also had acquired a modem company called Daycom, which was successful with PCMCIA modems, which was one of the successful things they did. They also made, using their modem skills, the modems that worked with the Series 3. We had modem skills in group as well as producing PCMCIA modems and they did the connection with the phones.

Weber: For instance the email client server. You wrote that in-house?

Davies: Yes.

Weber: So you certainly<*inaudible*>

Davies: Yes. We probably used Rand's IP stack. That's where he came in <*inaudible*>.

Weber: He had his own little company at the time.

Davies: He had his own little company called STNC which got bought by Microsoft for its mobile browsing and TCP/IP expertise. That's how I got to know Rand.

Weber: The browser he did, you never actually produced it commercially?

Davies: I don't think so, no.

Weber: Because that would have been <*inaudible*>.

Davies: In the end we went Opera on Series 5. Sometime later.

Weber: Late '90s.

Davies: Yes.

Weber: At the time when you were looking for this partnership of the mobile company, did you have an idea what form that would take? Something like what actually happened?

Davies: Well, there were several times in the history that we looked at partnerships. We tried to do a joint product with Motorola, for example, which was canned in the end, which was basically a quarter VGA [screen resolution] tablet device.

Weber: Oh, really?

Davies: Which we got very a long way with, and then they pulled the plug on that. We looked at buying phone companies, so we were looking, say one box or two boxes we were talking about and we thought one box would ultimately win. We saw the competition from the phone and thought we either have to be a phone company or partner with one. We chose to partner with one and that didn't go well and that's why Psion is out of consumer—I think Psion had to become a phone company like Handsfree. Handsfree did phones, did wireless. You need scale to become a phone company as a reality. It's what history has shown, and we didn't have the wireless skills. In the end, Psion Software sold to a phone company or sold the software platform as a smartphone platform and that later became a spinoff for Symbian. Now it's becoming Symbian Foundation as an open-source.

Weber: Interesting.

Davies: Yes, it's been quite a long ride.

Weber: I guess once you've looked toward phones in that sense, the people that made hardware would ultimately—

Davies: Yes, what happened at Psion in terms of history is that it sold its proprietary software, if you like, to Symbian which became a joint venture [in] which Psion had a stake in for some years and sold it out in the end and sold its stake to Nokia. But it stayed on and still exists as a company and also makes hardware. It doesn't develop its own software platforms anymore. It's actually Microsoft-based products largely.

Weber: What's your—

Davies: <*inaudible*>

Weber: For what?

Davies: Warehousing logistics.

Weber: Parking tickets.

Davies: Parking tickets, yes. Probably the biggest thing is supply chain logistics.

Weber: Supply chain?

Davies: Yes, a big customer would be UPS.

Weber: Oh so the UPS has a credit card—

Davies: Well no, somebody is tracking it when you sign one of those things; you do it on a screen.

Weber: Also, for instance, in restaurants, the credit card things.

Davies: Yes, all of those things. Integrated laser scanners, variants that work in a meat fridge with heated LCDs [liquid crystal display]. They're actually turned towards producing a phone now for the industrial market which has got a battery life that can really go the day with Wi-Fi and stuff. Ports, anything,g moving

bits of things from one of service to another. Volkswagen parts, that kind of stuff. That business retains but the platform it developed ended up in Symbian. That's ending up in the Symbian Foundation which is now the biggest open-source in the nation ever, I think. Massive.

Weber: In terms of value?

Davies: In terms of lines of code and accumulated R&D spent. Now the next phase is open-source.

Weber: When you talk about the strategy—these are clearly big directions. What was the dynamic? Was it David Potter that was—

Davies: I can remember the decision to license our software. It was a big decision and I was in his country home and it was basically the board of Psion meeting. The discussion was an intellectual debate, can you imagine? I remember the kind of arguments of, "Should we die like IBM or like Apple?" Apple was dying at that time, right? This is before Steve Jobs had come back to save the company and the reflection at that time was that Apple had made a mistake in not licensing its windows software, and this was before Apple had made some kind of a comeback. It's interesting to have a historical perspective but the view was, "Should we hang on to this proprietary software, or not?" We decided that we would license it to our competitors, not thinking that we would license it to mobile phone companies but to Sharp, for example, who was a competitor at that time. To HP, if they would take it, they were a competitor at the time. We thought we had something because DOS obviously wasn't the answer despite what people said. For a good quality user experience you needed something that stayed on all the time. Didn't need reboot. You didn't want to see C coming out all over. You wanted a graphical type of user experience in multi-tasking and overlapping windows and stuff like that. We thought we had something, and that's when we created the division; out of that strategy meeting. We created a division called Psion Software.Then the management team decided to target the mobile phone vendors as interesting customers to get revenues from because they were trying to get license revenues. I can remember my discussion and my journey to Finland to try and sell them the merits of this EPOC32 and frankly the Series 5—which is up and running as a prototype—was our selling card. It was just starting to sell we could show them an operating system and we displayed this for Nokia and Nokia thought well, they'd rather not be beholden to Microsoft and they'd rather have a counter to Microsoft. I think that was their strategic thinking at that time so they created Symbian. Symbian has since sold what, 250 million devices with Symbian operating systems. That's a fair number.

Weber: Not all Nokia phones, just the smartphones, right?

Davies: Not all Nokia phones, no. Sony, Ericsson phones, Japanese phones.

Weber: Only the upper end of the ranges?

Davies: Yes, smartphones. Quite a big range, you'd be surprised. Over time, as Moore's Law occurred, the line where you had hardware powerful enough to run Symbian OS or the multitasking OS has come down. When multimedia happened—it needed the proper elbow room—it was nothing compared to

capturing a jpeg. It's really multimedia that's moved the requirement of the memory up and the OS is nothing in that. The line on which you can run proper OS's is—as we've certainly done. The proportion of devices that have hardware strong enough to run OS's is in a increasing slice of total market. And will continue to produce.

Weber: You probably have more installations on phones than you ever did on [previous] devices.

Davies: Oh, way more. You know in a day we sell more than.

Weber: This critical strategy was mid-nineties, right?

Davies: Yes, I would say '94, something like that.

Weber: Was there any dissension?

Davies: No, it was consensual. No big battles and no resignation, a sober discussion of—

Weber: This is a board that knew each other well?

Davies: Yes.

Weber: So the board was mostly people who had always been involved.

Davies: People who'd been there a long time. Didn't have a high turnover of people.

Weber: Big board? Small?

Davies: I'd say seven people who got along with each other, definitely. They're friendly, personally friendly. Understood each other. No animosity. Good team spirit. Nobody with an axe to grind. No political.

Weber: A stable group.

Davies: A stable group that had been together a long time.

Weber: *<inaudible>*

Davies: I don't know. <*laughs*> No boardroom. Nothing like Scully, no. Nothing like kicking Jobs out. There was nothing like that. It wasn't dramatic. It was interesting and I knew at the time that it was a critical decision. Then in fairness, the team that set up Psion Software; all credit to them for saying, "Screw PDAs let's go for phones." That was pivotal. Two pivotal things; setting up and deciding to license and then deciding to go for phones. Then the opportunity presenting itself and being able to manage that.; something of a joint venture which was very exciting at the time which led to 250 units. Look, it kept Microsoft in its place. They basically would have been astonished that this small British company could stop Microsoft in phones. They were almost nowhere in phones. Still hanging around there with HTC and stuff. If you'd have said to them back then "This is what you're going to get for what you tried to give it your full," they wouldn't have believed it. Now we've got Apple and the iPhone. That's interesting. We should take that seriously and there's room and the Palm coming back with their new OS, so the story is not over. In other ten years' time we'll develop another ten years of history, interesting history of operating systems for mobile devices.

Weber: Does Samsung use some of your stuff?

Davies: Samsung uses some of our stuff, yes, Samsung, absolutely phones on Symbian. Very successful in Japan. Probably I'd say 20 to 30 million being shipped to Japan. Very advanced market. Very high standards. They wouldn't call them smartphones but they're smarter than most Western phones, frankly.

Weber: You'll support iMode, the Japanese?

Davies: Yes, FOMA [Freedom of Mobile Multimedia Access] is the 3G network, but iMode is the client We are able to export stuff to Japan. Very high standards. It has been very good for us in quality and things like that. If you can sell to Japan, you've got something going for you. They've been very good for us in terms of discipline and so very good spirit with the Japanese.

Weber: Easier to work with than the Finns?

Davies: Yes, the Finns are great. Scandinavians are just—you just have to take your time to get to know them. Japanese, also. You have to take your time. Relationships need time to build, but once you build them, and if you listen to them and understand them, they're very loyal. And very demanding. It's good for you. It's a very demanding market in Japan.

Weber: But similarly, it's a market where it takes a long time to get established, but then you have the relationship.

Davies: Yes, yes.

Weber: How much of the—I mean it is the same operating system, but how much of the applications development shifted towards more mobile things?

Davies: There's been a very interesting evolution of the OS. The OS had a very good scalable architecture. It's surviving Moore's Law because the client server model is very scalable and so all the system services are pretty much implemented by systems servers. That can handle large amounts of memory and lots of things going on. The thing that's really happened—first of all of course, Psion had an OS that didn't do phone—

<break in audio>

Davies: —things like signaling stacks had historically been done separately even if they're on the same core. They dock them with a separate phone—high end phone, you'd have a separate ARM 7 [microprocessor] running the signaling stack. Even if it's running on the same ARM as the application process, it tends to run in an isolated way that's not really an application on the application side.

Weber: It's a partial integration.

Davies: It's a partial integration but there's a reason to keep it at arm's length because signaling stacks have a very high IP content and are written to a [?]. Anyhow, the development of Symbian OS first had to relate in a multitasking sense, because lots of things want to access the phone. It's not just the phone app. You're doing email and using GPRS, you need to access SMS. There was the phone debt. There was moving to a Flash memory architecture—phones always use Flash memory. Persistent storage that didn't rely on a lithium battery—you never had a lithium battery on a phone, so that was different. The really big thing, I suppose, that's happened to phones is multimedia. Psion phones never played videos, never had cameras and that sort of thing. In terms of the development that's gone in, a lot of it is in the multimedia side. Symbian OS—where the growing sections are—probably the big bit is multimedia, and then—

Weber: What you were saying—making optimized performance and more efficient?

Davies: Just an awful lot of software. A lot of codecs [coder-decoder].—

Weber: <*inaudible*>

Davies: —and just when you open up a viewfinder, an awful lot is going on to that in terms of operating systems is quite challenging. Not to flat the battery straight away. If you look at the history of what's happened at Symbian, it's been phone integration. It's been on TCP/IP. It's been web browsers. These are the things that worked, so networking, GPRS. GPRS is quite challenging. It's like having multiple network cards because you could have *<inaudible>.*

Weber: You said what?

Davies: Multiple network cards? Do you know what multi-homing is? Multi-homing is when you have two network cards, so when you have a TCP/IP address you're simultaneously connected to two networks. A

router does that, but a PC normally does not. Phones, they might have a scalable number written so we had to deal with—

Weber: But why? Why is that put in a GPRS?

Davies: Every time you get a new PDP [Packet Data Protocol] context, that's like a new network card. You can connect to any network. That's just the air interface standard. It's done in such a way that you connect to it every time. You might have several PDP contexts going at the same time so you have to—we had to deal with multi-homing before PCs did. On an NT [Windows NT] machine it did multi-homing but we did it first. We had TCP/IP. We had the IPv6.. It never was used but the standard was there.

Weber: Why <*inaudible*>?

Davies: Well, I guess eight years ago and still don't use it. Do you use IPv6? It's disused. How often do you use it? It wasn't used for mobile phones so it was premature. Anyhow, loads of TCP/IP. Networking, multi-homing, multi-bearers, the ability to switch between Wi-Fi and multimedia, color screens. Increasing color depth, increasing resolution.

Weber: That's just <*inaudible*> more video now or not?

Davies: Well, it changes the software in a sense that the APIs that you've got to do and graphics accelerators now—you have a variety of hardware platforms. There's been similar evolution to what's gone on PCs. Remember when people called them "multimedia PCs?" You wouldn't say multimedia PC now, but there was a time when you would have said that.

Weber: So the Series 5 didn't do gaming on it.

Davies: It did do games but we only had built-in games with things like Solitaire. It didn't do twitch games and it didn't do Doom, stuff like that, but the phones do. Phones can do all of that. That's the development. Bluetooth has come. A lot of Bluetooth profiles.

Weber: The OS was already very generalized writing to different phones was <*inaudible*>.

Davies: Well, it's got a hardware adaptation on it. It's not like Wintel. Remember when Windows tried to run on MIPS and that never worked?

Weber: Right.

Davies: We've always had to deal with a variety of hardware platforms. PCs are easier to build because Microsoft actually controls the hardware specification. When you have USBs [Universal Serial Bus]—

Microsoft is highly controlled. For phones, because there are a variety of price points to hit and a variety of phone factors, the hardware adaptation is more challenging; particularly if you're going to talk about a range of devices and a range of price points.

Weber: <*inaudible*>

Davies: That's right. It is a lot and Symbian is definitely <*inaudible*> but that's<*inaudible*> choice.

Weber: But you could easily.

Davies: There are different versions of ARM architecture, and there's a wide range of different hardware capabilities and a lot of adaptation so there are many different hardware-software combinations. It's not like Wintel which is more—there's more uniformity in the hardware and PCs than there is in phones, let's put it that way.

Weber: But you could rewrite it for the processor?

Davies: You could, yes. It does take advantage of ARM. It's multi-tasking so it's very tuned to ARM.

Weber: Okay, so there is a type.

Davies: I think Windows is, in reality—

Weber: Hum?

Davies: I think Windows is in reality as well, tuned to—

Weber: Apple is kind of interesting that it switched—

Davies: That's right, that's right, and they kept that going for some time. They kept quiet about that. We have the ability—I mean kernel developments are sometimes done on x86 so it is possible to run EPOC in x86 if there's no commercial drive to it. Yes, you're absolutely right. Most people wouldn't know the difference. It would easily be portable <*inaudible*>*.* Does seem to have the right characteristics for better devices and it's clean, basically cleaned up.

Weber: The first version of EPOC was already out?

Davies: EPOC32 was on. EPOC16 was actually—

Weber: Right. The Series in the market: that was the end of the standard Organiser.

Davies: Series 5 did okay. They had decent sales. There was a Series 7 which was I think what the Netbook was called. That's where we called it a Netbook and you could put a Wi-Fi PCP card [?] and it had a kind of leather surround. It was smaller. That wasn't a commercial success. That was on. Even the so-called Netbooks you get now I consider just standby. You still have a need for something that you just open and use. One day we'll get there. Like a phone. You don't wait for a phone to boot.

<audio fades>

Weber: Any paths you didn't take that you wonder—?

Davies: I think they essentially let Palm in because we underestimated the significance of the PC, the importance of connecting to the PC. <*audio skip*> —for the modem business then turned up to have some value in the Palm business. I put that down to: we didn't see the wider PC ownership at the right time and that was a disadvantage of being in the UK. There's an argument as to whether we could have become a phone company but the history has shown—well, maybe we could have become a—we'll see what long-term happens. You need scale. One of the things we discovered is that the amount of R&D to build a phone compared to a PDA is almost an order of magnitude higher. Wireless and the amount and type of approval you have to do to making a phone and selling to carriers, operators and all—it's a serious undertaking. Maybe we should have had a go at that more. I'm not sure we would have been successful. Doing a joint venture with Motorola was a reasonable choice but that turned out not to work and Motorola had a downturn at that time, so no. I think people and companies come and go. I can't think where we would have—we probably could have milked it—extended the life. We probably got out slightly too early of the notebook market. Could have innovated and kept on going but we wouldn't have stopped phones from taking that over. The idea of having PDA functionality in a phone is totally compelling. Phones were the future. Having the software in the phone is probably the happy outcome.

END OF INTERVIEW

