An Interview with

JONATHAN SACHS

OH 388

Conducted by Martin Campbell-Kelly

on

7 May 2004

Needham, Massachusetts

Jonathan Sachs Interview

7 May 2004

Oral History 388

Abstract

Jonathan Sachs describes his personal and educational background and the various jobs he held at MIT, Data General and Concentric Data Systems. He tells how he started developing spreadsheet programs and how he met and started working for Mitch Kapor. He discusses the development of Lotus 1-2-3 and his role in Lotus Development. Finally, he describes the reasons he left Lotus and some of his activities since then.

Preface

As part of the Software History Center's collection and preservation activities, and in conjunction with its meeting on the history of personal computer software held in Needham, MA, on May 7, 2004, the Software History Center (SHC) arranged for 14 oral histories to be conducted with computer software company founders and other key industry participants. All of these oral history interviews were conducted by historians well qualified by their knowledge and interest in computing history.

The following is a list of the people who were interviewed together with the name of their interviewer:

John Brackett and Doug Ross, interviewed by Michael Mahoney
Dan Bricklin and Bob Frankston, interviewed by Martin Campbell-Kelly
Dan Bricklin and Bob Frankston, interviewed by Paul Ceruzzi
Jerry Dreyer, interviewed by Thomas Haigh
Ben Dyer, interviewed by Nathan Ensmenger
Dan Fylstra, interviewed by Thomas Haigh
Gary Harpst, interviewed by Tim Bergin
John Imlay, interviewed by Bill Aspray
Luanne Johnson, interviewed by Janet Abbate
John Landry, interviewed by David Grier
Mike Maples, interviewed by Nathan Ensmenger
Seymour Rubinstein, interviewed by Jeffrey Yost
Jonathan Sachs, interviewed by Martin Campbell-Kelly
Oscar Schachter, interviewed by Thomas Haigh

Each interview was tape recorded, transcribed and edited by the SHC, the interviewer and the interviewee to ensure clarity and readability without changing the style or flow. The original tapes along with the edited transcripts were donated by SHC to the Charles Babbage Institute (CBI), which placed the edited transcripts on the CBI website and have archived the audio tapes.

On January 1, 2005 the Software History Center merged with the Computer History Museum, and its work is continuing as the Software Business History Committee as part of the Museum's activities (see www.softwarehistory.org).

Software History Center Oral History Program
Jonathan Sachs Interview

Martin Campbell-Kelly: It's Thursday, May 7, 2004 and we're at the Sheraton Needham Hotel near Boston. I'm interviewing Jonathan Sachs. I'm Martin Campbell-Kelly from Warwick University in the UK. Jonathan, tell me a bit about your social and educational background: date of birth, parent's occupation, siblings, and schooling.

BACKGROUND

Jonathan Sachs: I was born June 25th, 1947 and was an only child. I went to MIT, entered in 1964 and graduated in 1970 after a brief dropping out in 1968, and then returning later to finish my degree in math.

Campbell-Kelly: Tell me about your parents and where you were born.

Sachs: I was born in Baltimore, but we lived in Rhode Island pretty much the whole time until I went away to school, with the exception of a year in London and some summers abroad. My father was a professor at Brown University. His field was Babylonian astronomy and he read, transcribed and translated cuneiform tablets that were mostly concerned with astronomical observations. So he was very attracted to the British Museum in London which has the world's largest collection of tablets.

My mother didn't really have a career in the same sense. She did work for *Mathematical Reviews* part time as a proof reader for a number of years and as an office manager. She worked in the department that my father was in -- it was a two-person department-- and she did secretarial things and helped writing grants and things like that.

Campbell-Kelly: You started at MIT in Sept. 1964?

Sachs: Yes.

Campbell-Kelly: What courses did you take and what were your computer experiences as an undergraduate?

Sachs: The summer between high school and college and the summer after that, I was really fortunate to get a job at the Woods Hole Oceanographic Institution. When I was in high school, we had a very good chemistry teacher who had connections with Woods Hole. Woods Hole also had a program where students could come up and get lectures periodically on different topics. There was an oceanography club that I was a member of. I had access to the Brown library through my father's card, and I used to wander through the science library. One day, I was reading an optics journal with an article by a guy who had done an analysis of how ocean waves focus light. It was a very crude mathematical analysis. I was just learning calculus, and it actually happened to be a problem that you could solve with high school level calculus. I got really interested in that and I developed the equations for how that worked. A friend of mine who was also in the oceanography club managed to talk some a researcher named Ray Stevens at Woods Hole into hiring us for the summer, and this fellow had a whole research lab that was studying ocean waves. They had a wave tower out in Buzzard's Bay and telemetry equipment for sending information back. I actually got to do original research for a couple of summers.

Campbell-Kelly: How was that relevant to computing?

Sachs: The second summer I did get to use the computers and write programs. Some of the data we were collecting was analyzed using the FORTRAN language. Basically I used FORTRAN for quite a while -- it was the only thing I knew for quite a number of years, so mostly I was a scientific programmer.

Campbell-Kelly: When you returned to MIT, what did you graduate in?

Sachs: I graduated in mathematics. I went in thinking I was going to major in chemistry and skipped the first year based on my high school work. So I went directly into organic chemistry as a freshman and that definitely convinced me that chemistry was not something I was meant to spend more time doing. Math was kind of my next best skill, so I switched majors.

Campbell-Kelly: When did you graduate?

Sachs: I graduated in 1970.

Campbell-Kelly: Then you stayed on as a programmer. You were there for about ten years at MIT?

Sachs: Yes.

PROJECTS AT MIT

Campbell-Kelly: Talk about some of the jobs and the kind of things you were doing. You were at the Center for Space Research, I think?

Sachs: Yes, I had a number of part time jobs while I was a student, and then after working at Woods Hole, I worked a couple summers at the Jet Propulsion Lab in Pasadena analyzing magnetometer data from the Mariner IV satellite. We used FORTRAN.

Campbell-Kelly: And you were at the Cognitive Information Processing Group?

Sachs: Well, my connection with the Jet Propulsion Lab got me into the space side of it, so when I dropped out of school and needed work, I was able to use that experience to help me get a job at the Center for Space Research at MIT. They were analyzing some plasma data from an earth-orbiting satellite. Then after that I left Boston briefly and moved to San Francisco to study Zen, and couldn't get a job there. I eventually came back and got a job at MIT through a friend of mine. That was actually a big turning point because up until then, everything had been FORTRAN and entering programs on punch cards. You would submit a deck of cards and if you were lucky, maybe twice a day you would get things run on the computer. The compiler would tell you that you made some minor mistake and then you would go back and correct things; this was very laborious, and very slow. But they were using minicomputers and had a PDP-9 [manufactured by Digital Equipment Corp.], which was a giant, room-filling behemoth. You signed up to use it for like an hour or two during the day. They were building a reading machine for the blind and it had a Vidicon scanner and speech synthesizer. I was working on some of the character recognition part, and you had a scanned image of the page and had to convert that to a stream of ASCII text.

In addition, it was just sort of an electric moment the first time I sat down in front of a computer and I typed something and it typed something back! I feel sorry for people who didn't get into computers earlier, because they take it totally for granted as if it's nothing. But it's positively electric, especially when this thing is waiting for you --it's doing nothing but waiting for you --there's enormous pressure to just do something, you know? It was a very expensive and very limited resource.

Campbell-Kelly: You developed a program system called STOIC (STack Oriented Interactive Compiler?

Sachs: Yes, that was for the Biomedical Engineering Center. That was actually after another job at the Center for Space Research. When that ended, I got a job working with a group that was building biomedical instrumentations around microprocessors. So they had four different instruments: there was a heart monitor, an inner-ear tester and a couple of other things. But they needed a programming language to write the software. I got interested in Forth. Some people say all programmers are frustrated language designers. I certainly have written my share of programming languages over the years. I had written several variations of Forth. This was probably the third one that I got to play with. It was very similar to Forth; it was very cleaned up in certain ways.

Campbell-Kelly: Did that have any influence on designing spreadsheet programs?

Sachs: Writing compilers was relevant because there's a compiler embedded in every spreadsheet that handles the expressions that you type in. So I definitely knew how to do a language, and adapt part of it based on some techniques that I had developed for various projects.

Campbell-Kelly: About 1978 you left MIT to go to Data General?

Sachs: Yes, the summer of 1977, I left MIT. I'd been working there for a long time and I'd written a lot of stuff that was kind of interesting, but the problem with working at MIT is - they get money for a grant; they do something and then they do something else. There's never a wider audience for this stuff. So I began envying people working in industry who wrote the tools that I used -- and where they would write something a lot of other people would use. And what I also got out of my system later on was the notion that I wanted to get more into management and run larger projects. That was also something I never really had the opportunity to do at MIT.

MINICOMPUTER PROJECTS

So I did interview at a few places but accepted a job at Data General. I started working for a guy named John Henderson, who was one of the early employees -- #14. It was in a group called Small Business Systems. They had a product that they sold which was an interactive COBOL (Common Business Oriented Language) system. They were interested in developing a new operating system that was a friendlier platform for their system to work on. So he hired me to start working on that, and we in turn hired a team of people -- one of whom, incidentally, was Ray Ozzie. So John Henderson actually hired him out of the University of Illinois, his alma mater. So that was kind of an accident. And it was a lot of fun working with Ray - he's a very bright guy.

Campbell-Kelly: You formed a partnership with John Henderson?

Sachs: Right. That was somewhere around the beginning of 1980. He left Data General where he worked for a guy named Larry Seligman and had worked for him for years, and absolutely couldn't stand him and finally said: "I can't take it anymore; I'm leaving." So he started a consulting company, and prior to developing the COBOL system at Small Business Systems, they had developed a point of sale supermarket checkout system for a cash register company called Sweda. So he was very familiar with how that system worked, and he went back to Sweda and they actually paid him to make enhancements to the system. So I joined him about six months after he left, and we were doing contract programming. While we were there, a fellow named Allen Klutchman, who was still at Data General but was planning to leave, approached us to write a VisiCalc clone for Data General computers.

Campbell-Kelly: What kind of computer was that?

Sachs: It was for mini computers, although we did an Eclipse version later for AOS Ultimately, he was thinking it would be popular because of a product that was coming out called the Micronova, which in fact didn't really ever go anywhere. So that was my first exposure to spreadsheets.

FIRST SPREADSHEET PROJECTS

Campbell-Kelly: What did you know about spreadsheets?

Sachs: Prior to that I'd never even heard of them. Although interestingly, while I was at Data General, Bob Frankston and Dan Bricklin had come back and done a demo of VisiCalc to our group and they had sort of written it off as uninteresting. So they (Klutchman) loaned us an Apple II and they gave me a VisiCalc manual. It struck me as an interesting programming challenge because it incorporated a lot of things that I had done. The more I worked on it, the more I came to appreciate that it was a very powerful idea.

Campbell-Kelly: Had you met Dan or Bob before this time?

Sachs: As it happened, I met Dan some years before at Software Arts because I was living with some other people in Somerville and one of them was his girlfriend -- who's now his wife. So he used to come by every once in a while, and I guess he was working on word processors at DEC (Digital Equipment Corporation) back then. So I knew him, but I didn't really know Bob or Dan in their capacity at Software Arts.

Campbell-Kelly: So you finished the program?

Sachs: Yeah -- then Klutchman's company which was called Access Technology handled the sales and marketing. Allen Klutchman tragically committed suicide a number of years ago after his wife died. But they were able to sell some copies of our software; we got some minimal royalties, and they came back to us and wanted us to write a version for DEC mini computers.

Campbell-Kelly: What was it called?

Sachs: *Supercomp* I think.

Campbell-Kelly: This was on the mini?

Sachs: Yes, it ran on Data General hardware. I wrote a version for DG minicomputers and Access Technology wanted an additional version to run on DEC computers – they ended up paying someone else to write the DEC version. It was very similar to VisiCalc, but there were some differences.

Campbell-Kelly: How did you reverse engineer it?

Sachs: I just went from the manual and said: “Okay, I can write something that does this.”

Campbell-Kelly: And did you copy the same commands and screen layouts?

Sachs: I tried to improve it in various ways. It wasn't a complete copy, but functionally it did all the same things. It didn't take all that long. It's not that complicated a program.

Campbell-Kelly: What language is it in?

Sachs: The first version was written in Data General assembly language, and I subsequently re-wrote it in a higher level language called BCL. [BCL was a variant of BCPL (a predecessor of the C programming language) and we purchased a copy of the compiler from Bill Henke who lives in Belmont and who developed and maintained it.]

Campbell-Kelly: This is at Data General?

Sachs: No, this was at Concentric Data Systems [the partnership with John Henderson]. We were kind of toying with the idea of doing a version for some of the microprocessors -- the CP/M machines that were starting to come out. So it seemed to make sense to do it in a high level language.

Campbell-Kelly: And you'd retained the rights to this program, even though you eventually did it for other clients?

Sachs: Yes, we owned the source code, and they [Access Technology] had the marketing rights for the Data General version, so that was very clear. Then the relationship with Allen Klutchman deteriorated rapidly. He and Henderson really didn't get along very well. So we refused to do the DEC version and he hired someone named Faschetti to write it - I forget his first name. By that point I was really kind of intrigued with spreadsheets and I wanted to go on working on them, but it wasn't bringing in any cash. And Henderson was still contracting and making money, so that put a lot of strain on our business relationship. We decided to go our separate ways somewhat. I still play bridge with him every month and we're actually good friends.

We had equal stock ownership in the company and we agreed to split the technology. So I took away with me the rights to continue to develop spreadsheets, based on this high level [BCL] language version of the spreadsheet that I had, and he kept a copy of everything that we had, as well.

At some point, I don't really remember exactly when, Mitch Kapor had visited us in our offices, which were in a basement in Cambridge [180 Franklin St. in Cambridgeport].

Campbell-Kelly: Why was that?

WORKING FOR MITCH KAPOR

Sachs: I'm not sure how he [Kapor] knew Henderson, but he definitely knew of the spreadsheet work we had done. He was sort of exploring if there was some way we could work together. He was interested in spreadsheets. That never really went anywhere, but it sort of stuck in my mind. So when the relationship with Henderson broke down, that was one of the places I decided to explore. I did meet with Mitch sometime in late 1981 - summer of 1981 maybe. What we had in common was this real enthusiasm for spreadsheets and a desire to push the metaphor a lot further. He had done some work for a company called Personal Software [the marketing arm for Software Arts] and he had a couple of products that extended VisiCalc called VisiPlot and VisiTrend. He had a revenue stream from those royalties that he used to start a company called Micro Finance Systems. I'm not quite sure how long it had been in existence, but not for that long.

So I visited him at the offices there at 180 Franklin St. in Cambridge. We had a chat, and frankly I said [to myself]: "This is a real long shot, but what the heck; he'll pay me to keep working on what I'm interested in and if something works out, fine, and if not - then I'll do something else." So we started working together using the version of the spreadsheet that I had as a base. He had some very specific things he wanted to make sure we could handle, like variable column width -- the original version of VisiCalc required every column to be the same width. I said fine, no problem, and I added that. I rewrote it in the C language.

Campbell-Kelly: Was it still running on a mini?

Sachs: No, this was on an Apple II with a CP/M card and the compiler was called BDS C which stood for Brain Damaged Systems C! It was actually a pretty good C compiler - it was adequate to get the job done. It [the prototype spreadsheet program I developed with Kapor] wasn't a commercial product, but it was certainly enough of a prototype that it was sort of a tool that we used to play with the algorithms and experiment with different ideas. So we were in kind of a holding pattern with that. Then I remember one day this IBM PC showed up in the office and everyone was kind of standing around staring at it.

Campbell-Kelly: Where did that come from?

BUILDING LOTUS 1-2-3

Sachs: I know Mitch got it from somewhere. He had a lot of connections with computer stores, so I think someone at some computer store must have called him and said they'd got one in and asked if he wanted to check it out. I don't know if he bought it or had it on loan or what. He got one of the very first ones.

I think we both really saw the potential in it -- that it was likely to be a good platform to target. It had the IBM name - which really legitimized it a lot more for business users, and the Intel architecture. It was a lot more robust than the Apple architecture in terms of supporting large amounts of RAM. So after a very short decision period we said: "Let's go for this one." I started re-coding the C version of the software in Intel 8088 assembly language and getting it to work on the IBM Personal Computer.

Campbell-Kelly: Did you do a comparative analysis of the products on the market like Context MBA for example?

Sachs: Context MBA wasn't on the market at that time. There were very few other spreadsheets.

Campbell-Kelly: Integrated packages were up in the air a bit?

Sachs: No, nothing. There was something called SuperCalc which was a competing spreadsheet and you could count them on the fingers of one hand. Context MBA didn't come out until a few months before 1-2-3 came out, because I remember seeing the manual for it when we were 80% through the development.

But Mitch's background was adding graphics to a spreadsheet. It was pretty clear to him that graphics added a lot to the spreadsheet and it was kind of obvious to me. I'd done a lot of graphing software at the Center for Space Research, and I knew exactly how to write the routines for scaling axes and drawing graphs. That came fairly easily for me.

Then we were sort of casting around for other things to integrate with the spreadsheet. Originally the third component of 1-2-3 was going to be a word processor. I started working on that after the graphics part was done. One thing I should mention is the way I work. It's a lot easier to do this when you're doing a one person project, but relatively soon after starting there was a working prototype. There were a lot of functions missing, but it was useable. So maybe you could just type in numbers but not formulas, but I ended up working off-site. It was too noisy to work in Cambridge, and I was living a distance away and it was a long drive back and forth. So I would basically come in once a week with a new version and meet with Mitch for an hour or two in the morning and then go back. During that week, people would be using the most recent version; if they found problems they'd call me up and I'd fix them for the next week. I always gave the highest priority to fixing bugs, so at each point in time there was a whole series of these versions -- each of which worked. So while the later features were being added, the former ones were being debugged and refined. This was much more efficient than the textbook method of designing and writing the entire program first and then testing and refining it later.

The first thing to come together was the spreadsheet part and then the graphing part. I was a few weeks into working on the word processing part, and I was getting bogged down. That's about when *Context MBA* came out, and I got a look at what they had done. They had five integrated components, and the other two were communications and database. I hadn't really thought about the database angle, but when I saw that, it just seemed like it was a natural addition. It had better synergy with the spreadsheet than the word processor did, and it would be a heck of a lot easier to implement.

So I didn't have too much trouble talking Mitch into switching to that as the third part, and it was really just a few weeks of work to do most of the database piece.

Campbell-Kelly: Did you make that compatible with existing database formats?

Sachs: No. It wasn't really a separate database; it was a way to treat a range within the spreadsheet as a database. So there was no separate database file. Everything was in the spreadsheet file [in memory]. It was only for very small databases, but it was still quite useful for a lot of smaller size data analysis problems.

LOTUS 1-2-3 FEATURES

Campbell-Kelly: There are quite a number of significant innovations in *Lotus 1-2-3* compared with *VisiCalc*. Could you say a few words about each of them and identify the prime mover of these things - particularly for the user interface and command structure.

Sachs: Mitch was the primary driving force for user interface issues. He had invented the so-called *moving cursor menu* which we incorporated. He did that for VisiPlot. When you arrowed right and left in the menu, it highlighted the so-called short prompts and displayed a long prompt that explained what each thing did.

There were certain features that he insisted on being there. A lot of the naming of the menu items was done by committee, for example, there are constraints that you have to have unique first letters. I remember there was a conflict between graph and global that we had a lot of meetings about. I don't remember how we ended up resolving that.

As far as individual features -- certainly at the lower level of the implementation, Mitch pretty much left me alone. We did have these weekly meetings, so we stayed very much in tune, and we would discuss things and he'd throw out ideas and I'd throw out ideas. Whose idea each individual thing was is something that's not clear.

Campbell-Kelly: Were you developing in assembly or in C?

Sachs: This was all in the IBM PC's Intel 8088 assembly language. I had done a preliminary evaluation. There were only one or two early C compilers when I started, and I did evaluate one of them. I took one of the modules and I compiled it in C and then I compared that to writing the code by hand. There was easily a factor of 5 times difference in size and speed, so it just didn't seem like that was in the cards. [A C compiler generates assembly language automatically – I compared the compiler-generated code with my hand-generated code to see what the penalty would be for using the compiler. Writing in C is much easier and faster than writing in assembly language but at that time the resulting programs were much slower and larger.]

I wasn't really afraid of assembly language; I'd been writing it for years and years. But it's a lot harder than writing in C. The source code for *Lotus 1-2-3* was a stack of paper about 8-10 inches high. That was a lot of writing.

Campbell-Kelly: You mentioned you'd write on the screen directly to get a speed improvement.

Sachs: Yes, spreadsheets are extremely screen-intensive. There's a lot of updating of the screen. Every time you scroll you're repainting the entire screen, and any inefficiency in that area would really be very noticeable in the performance. The IBM PC had a very elegant text display and it had very high quality characters: green on black, and it had a very efficient interface that mapped the screen into memory. If you wrote directly into the memory buffer for the screen, it would take microseconds to update an individual character. So basically *Lotus 1-2-3* wrote right into the screen buffer when it needed to update the screen.

Most of the other functions: the file I/O, using the keyboard, and so on, all used standard BIOS calls, but for screen I/O we definitely bypassed the BIOS.

Campbell-Kelly: What about the macro language?

Sachs: That was kind of an afterthought at the end. We recognized that having some way to automate operations and do some rudimentary programming would be very valuable. The macro language from an implementation standpoint is extremely easy to write compared to a real language that has variables and expressions and so on.

It was kind of an expedient addition. We never really imagined that it would catch on quite the way it did. But in defense of macros, it's something that non-programmers typically found non-intimidating compared to actually having to sit down and learn a programming language. Once you knew how to use the program, and knew what key strokes to type, you just entered those key strokes in and it would do it. So the simplicity of it was very appealing to our customers who for the most part were non-programmers.

The dark side of macros is that they're tied to the exact menu system and it sort of made it impossible to change anything -- it greatly constrained what you could change without becoming incompatible with pre-existing macros. And later as people developed very elaborate macros, it became a problem.

Campbell-Kelly: There's a phrase: eating your own dog food. The idea is that the programmers actually use the system that they're developing as they develop the system. Is that what you did?

Sachs: Yes, actually about half way through the project, when there was a useable spreadsheet, I started using it to maintain the feature list or the features as yet to be implemented list. Mitch and I had a long list of things that we would have liked to implement. I kind of made an estimate of approximately how long I thought it would take to write each one, and then we assigned priorities to each feature.

Basically you could sort the list by priority and total how many days of work there were left to do, and compare that to the release date. That gave you a cutoff as to what was going to be in and what wasn't. That worked very well actually. That was kind of our project management tool. But because of the iterative development and always having something that worked that was mostly debugged, it sort of allowed us to stop at any point in time -- whereas if we'd started with a functional specification and committed to that, it would have been a lot harder to control the release date.

Campbell-Kelly: Did you keep any kind of documentation as you went along, or was it just all comments in the code?

Sachs: Pretty much just comments in the code. Mitch kept detailed design notes which I suspect he still has a lot of. I was too busy writing the software to do anything.

Campbell-Kelly: Lotus also had a nice online help and tutorial systems. Was that part of the original release?

Sachs: Yes.

Campbell-Kelly: Did you do those?

Sachs: I wrote the infrastructure for the help system [this consisted of a help compiler which was the help authoring tool and the help engine which rendered the help text on the screen at runtime] and then several people in the office wrote the actual help text itself [i.e. the textual content of the help system]. I know John Posner was involved. I wasn't completely sure who was doing what because I basically had my nose to the grindstone.

Campbell-Kelly: They were doing other stuff as well?

Sachs: Well, they were writing the manual and then there was a whole company that was evolving-- kind of in my absence -- where they had to be able to do manufacturing, marketing, sales, tech support, PR, human resources, warehousing, payroll, building management, administration, etc.

Campbell-Kelly: About how many instructions were there in *Lotus 1-2-3* release 1?

Sachs: I don't know about instructions; the program was about 85 kilobytes which is incredibly small by today's standards. However, by my standards, I'd been working for years on minicomputers where there was a hard limit of 64Kb which included both the application and the operating system. Because the first IBM PC's had a 64Kb memory limitation, there was a big psychological barrier for me when the program got nearer and eventually exceeded 64K. Mitch kept reassuring me: "Don't worry about it."

Campbell-Kelly: So it must have been something like 25,000 instructions?

Sachs: I have no idea. I don't know how many bytes the average instruction is - probably 4. 64,000 / 4 = 16,000 instructions.

Campbell-Kelly: My understanding is that Mitch originally offered *Lotus 1-2-3* to Personal Software. Do you know anything about that?

Sachs: I never heard of that. I'm a little skeptical about that, but it's possible. You'd have to ask Mitch. We did offer the software to IBM at one point.

LOTUS DEVELOPMENT AS A BUSINESS

Campbell-Kelly: Were you involved in the raising of venture capital for Lotus Development Corp? Were you essentially an employee or were you an equal partner?

Sachs: I had equity, and they considered me a key employee. I remember I was peripherally involved in raising funds; I certainly met Ben Rosen and L.J. Sevin who were the main partners. They insisted on both of us having doctors come by and examine us to make sure we would survive. I remember one amazing incident. For some reason, the four of us were standing around and the question, "What's the most expensive dinner you ever had?" came up. L.J. Sevin -- he made a lot of money from Mostek I think --said: Well, I went to Paris once for dinner." So I think that sort of trumped everybody.

Campbell-Kelly: How was the equity shared between Kapor and you and some other people?

Sachs: I forget exactly how much Mitch had. It originally started that he had something over half -- I don't remember exactly what number. I basically had 15%. This was pre-investors. There was the chunk that was left un-allocated for an "as yet to be hired" sort of VP of marketing and sales, and then there was a chunk that was left unallocated for an employee pool. Of course, that all got diluted when we got venture capital, so I think I was down to like 4 or 5% after that.

Campbell-Kelly: How many employees were there at this stage?

Sachs: I would have to guess about 40 to 50 by then. There was a lot of hiring going on at that time. The number of employees was quite small during the first half of the development and then things got serious. There was a lot of hiring after the first Comdex, when the product was shown to the public around Thanksgiving of November 1982.

Campbell-Kelly: What were your relationships with IBM?

Sachs: I don't know about Personal Software, but we did actually offer *1-2-3* to IBM. Mitch and I flew down to Boca Raton -- I know it was the middle of summer, because it was beastly hot and I was wearing a 3-piece suit.

We showed them the software. They had introduced the IBM PC and they sold a number of programs under their brand. The timing was such that they wanted more software to help sell their computers, but they had recently sold a word processor called *EasyWriter* that was written in Forth and it was very buggy. They got a lot of bad press about that. IBM being IBM basically set rules that they weren't going to release any more software unless it was tested very thoroughly -- like six months worth of testing.

So I think the people at IBM -- when they ultimately decided not to sell *1-2-3* -- were doing it as much for our benefit as for theirs. They wanted to see it come out sooner but were afraid that if they got involved with it, the bureaucracy of IBM would sort of drown it. That decision turned out to be fortunate for everybody.

Campbell-Kelly: Tell me what the reception was like when you announced it at Comdex. There would have been other exhibitors at the same time.

Sachs: There were kind of three phases by which it became public. There was an article in the *Wall Street Journal* a few months before Comdex [written by Ben Rosen]. There was an official introduction at the World Trade Center. I can't really place when that was.

Campbell-Kelly: That would be January 1983, wouldn't it?

Sachs: Yes, so that must have been after Comdex. Also there was an extensive *beta* program and there were dealer kits. So somehow they had managed to generate a huge amount of interest in the product by Comdex. It wasn't that people discovered it then; the fish were already jumping out of the water by then. I know they booked about 3 million dollars worth of orders at the show. They did a lot of deals with distributors. I was doing booth duty. I gave demonstrations till I was hoarse. It was very exciting. There was a lot of interest.

USER FEEDBACK

Campbell-Kelly: Tell me about the difference between *Lotus release 1* and *release 1A.*

Sachs: One more thing about Comdex; it was very interesting because I think we were maybe 20 to 30 people by the time we went to Comdex, and the mood --the morale -- was very guarded prior to that. We weren't sure if there would be competition. We had no idea what else was going to be out there. We didn't really have any idea if the company was going to succeed or not. Then after Comdex, there was absolutely no question. Everyone was kind of re-energized. So it made a huge difference to the way it felt.

Once the product was released, that was Version 1. Version 1 ran only on the IBM PC and the Compaq luggable computer which was another one of Sevin/Rosen's investments. They kind of hit two home runs in one season. As soon as it [1-2-3 Version 1] was released, we were approached by a lot of other companies who had other models of PC and they wanted us to support them.

The first thing I did was to isolate those parts of the code that were dependent on the hardware, i.e., that had to do with the screen and the keyboard. A couple of things I remember -- there were four drivers I think. The other things had to do with printing and printing graphs.

I isolated those so they could be dynamically loaded and I wrote a set of reference drivers for the IBM PC, so people could see how that worked. Basically at that point, they had hired managers who hired programmers whose jobs it was to do individual porting of versions. So Version 1A had no new functionality; although I think there were some very minor bugs that were fixed and support for a driver -- so it could be modified. 1A* - I don't remember; I think it fixed some minor problem with the calendar.

Campbell-Kelly: Did you ever meet any of the users?

Sachs: I did meet with a few of the early *beta* users. There were a couple of them who were very sharp.

Campbell-Kelly: Did you make changes in response to their user feedback?

Sachs: Yes -- I don't remember what, but they did make some good suggestions. There was a guy named Allen Sharf who I think was at Merrill Lynch, and there was somebody else... Marv Goldschmidt, who was in charge of the *beta* program. I'm sure he would remember the names of more of the testers. But he also sort of channeled feedback to me.

But by the time the beta testers saw it, it was pretty near to being done. Also, one thing I insisted on -- based on my own experience at Data General and elsewhere -- was a 3 month kind of no-new-features period at the end of the development cycle, to let any bugs and other things shake out. As it happened, the last serious bug was found right at the very end of that 3 month period, so that turned out to be good because when you sell a million of something and you don't have an Internet to send out patches -- it is a total disaster to have a serious problem. As it happened, the software was extremely reliable.

Campbell-Kelly: How was the manufacturing? Was it subcontracted?

Sachs: No, they did it in-house. I guess for a time they contracted out the reproduction of the floppy disks, and then they eventually brought that in-house. The manuals were printed and assembled in a warehouse. There was a delay, a worldwide shortage of D rings which are the rings that hold the manuals in the binders, and that caused some delays.

Campbell-Kelly: Did the documentation need to be typeset? How was that done?

Sachs: Basically the writers did the typesetting themselves. There was someone [Barbara Kassabian] who did the blue boards and the layout, so that was all done internally. Our people were very experienced at it.

FOLLOW-ON PRODUCTS

Campbell-Kelly: Let's move on to *Jazz* and *Symphony*. Which of those were you involved in?

Sachs: I knew the *Jazz* team, but I had virtually nothing at all to do with it, and I was involved in *Symphony* for 3 or 4 months near the end of my stay at Lotus.

Campbell-Kelly: My understanding is that once *1-2-3* was up running and a successful product, Mitch Kapor thought that the next product would be *Symphony* and that it would perhaps replace *1-2-3* -- that was the plan.

Sachs: I think the general thinking at the time was *1-2-3* is good; 4-5 would be better. So there was a press to add more features, and there was also all the things that didn't make it that were on the original feature list after the cutoff. So there were extensions to the database, having multiple windows, and a bunch of other things.

It sort of became the dumping ground for all the new features and the adding on of word processing and communications which in hindsight had some value in terms of synergy with a spreadsheet, but no one in their right mind would use a spreadsheet as their primary word processor -- it's just not powerful enough. What really was unfortunate was that by adding so many features, it overloaded the menu tree so the menus got very deep -- you'd have to go through many layers to get to an individual command. It was just more than people needed. In fact, probably most people didn't need all of what was in *1-2-3.* I use *Excel* now, and I only use one percent of what's in it. So I definitely started to get a sense part way through, that this was collapsing under its own weight, but it had a momentum of its own at that point. They just kept throwing things in.

Campbell-Kelly: Tell me about the programming hierarchy at this point. You were Vice President of Programming?

Sachs: No. Actually when I interviewed with Mitch right at the beginning I told him -- I don't want to be VP of Software Development. I had some taste of managing even a relatively small team at Data General, and I found it really frustrating because I like to write software and then I'm competing with the people who are working for me. They're having all the fun and I'm resenting it.

Campbell-Kelly: But were you VP of Research?

Sachs: Yeah, Research and Development -- which was basically an artificial title just so that when I called people outside the company I'd have some credibility. But it's not as though I was in charge of a big R&D department.

Campbell-Kelly: Tell me about the process for developing software. How many individuals were involved; what was your involvement?

Sachs: I'll answer your previous question first. What happened was -- once *1-2-3* shipped, then all of a sudden there were a lot of questions about what we'd do next -- how were we going to make all these other versions. So Mitch, true to his promise, did hire two other managers from Software Arts. He hired Dave McElfresh and Del Troppito, who had worked together there. They actually set up a real software development department and Q&A. There were some systemic problems like management was going out and cutting deals to do versions of *1-2-3* for other machines, but without first ascertaining how difficult that work would be. So sometimes they committed to things that weren't technically feasible.

So there was some pushback on that and they finally established a procedure where there was a technical evaluation first. It was sort of crazy. They had to make some space; they set up an organization and wanted to translate languages and hired people to write drivers. They hired several groups to write add-ins. They were going to internally develop an Add-in Manager and several other things that were done. They could add in a new thing with *Symphony* where people could write code that was kind of like their own driver, but it would add extra functionality.

Campbell-Kelly: That was something that was first published with *Symphony*?

Sachs: Yes.

Campbell-Kelly: There was a development kit, wasn't there -- supplied for *Symphony*?

Sachs: Yes, and if you think the macro language constrained their future development, the development kit did even more so! They basically published a load map with the name and location every single internal symbol in the program. And people started using them, so that now meant that you couldn't change any of the internal subroutines in the program. That basically paralyzed all future development, because of that shortcut. They got some mileage out of it at the beginning, but that was a decision that -- if I had still been there -- I would have resisted.

Campbell-Kelly: Initially Lotus was not too fond of adding products. Is that correct?

Sachs: No, I think they wanted to encourage them. They had a whole department that was dedicated to helping people with add-ins. That was after my time; I'm a little fuzzy on that.

Campbell-Kelly: Had *Symphony* gone to market by the time you left?

Sachs: I think so.

Campbell-Kelly: What can you tell me about its market reception?

Sachs: It did very well in Europe because *1-2-3* never got translated into other languages until much later. So it was the first thing that a lot of people used, and it continued to outsell *1-2-3* in a number of European countries. But it was not particularly successful in the U.S., partly because it wasn't macro-compatible -- in that all the menus had to be changed. I think partly it was just perceived as too big and too clumsy.

So at some point after I was gone, there was a sudden awakening within the company that we had to do something. So they went back and did a *release 2* of *1-2-3* where they pulled together some of the people that had been on the *Symphony* team and they were able to add features to *version 1* and arrange it so it could be translated and so on.

LEAVING LOTUS

Campbell-Kelly: Tell me about when you decided to walk away from Lotus Development Corporation.

Sachs: I had a somewhat uncomfortable relationship with Lotus, really from the beginning. When I worked there at the beginning, I was the only one working on *1-2-3*. They had some other products for the Apple II that other people were working on, so I was kind of isolated and I was really the only professional programmer there. It was more of a sales and marketing company. There were two other programmers there before me: one was 12 and one was 13! They were doing some of the Apple II development.

Then I was working off-site a lot, because the physical environment wasn't conducive to concentrating, so I sort of lost touch with the rest of the company. I knew the new people but I think I was something of a cipher, especially during the heavy development period--about ten months of solid programming between when we decided to target the new IBM PC and when we got to the beginning of the bug-fix-only period. I was really almost totally out of touch and Mitch was building a whole company, so I wasn't really part of its culture. Mitch and I have very different personalities, and he used to be a DJ and he'd wear Hawaiian shirts and was very expansive. He had some brilliant ideas, but he just threw out a lot of ideas -- some of them weren't that good, but some of them were excellent. I remember at a later meeting after *1-2-3*, I was working on a word processor called *Manuscript* and I was meeting with him and a few other people. He had some idea and I wrote something down, and he said: "I can tell when I have a good idea because Jon actually wrote it down."

But I think it was hard for him to work with me because if I don't really see how to do something -- even if something sounds like a good idea -- I sort of foresee all the problems with it. It's kind of a dampening effect on someone whose enthusiasm is just to come up with this new idea. I sort of would push back on him and say “I don't think that's going to work” or whatever. Later he did a lot of projects with other people and by then he was sort of a god and nobody ever pushed back on him. They just tried to do everything he said.

Campbell-Kelly: What other kind of consulting activities did you do with Lotus?

Sachs: The reason I left -- as I said, I had an uncomfortable relationship with the company and when *1-2-3* became a big success, it really became clear that I had two sort of unpleasant choices: (1) I could stay and try and be VP of Software Development -- which I knew I didn't want to do, or (2) I could stay at the lower levels of the organization tree and work for somebody else. That would be extremely weird! So there just really wasn't a place for me in the organization at all, and I was really fairly happy to get out. They made special arrangements for me to be able to sell my stock earlier than I would otherwise have been constrained to, which actually cost me a lot of money.

But I was very nervous about all of my net worth being in one stock and I didn't really have a lot of faith in the company doing well long term. So I was eager to diversify.

OTHER PROJECTS

Anyway, when I left, I sort of fooled around with some other ideas, and there was a program called *ThinkTank* that had been released by Dave Winer which was an outlining program. I was a little taken with that and I started playing around with that. I wrote something that was similar and then I starting adding more word processing features to it and eventually I approached Mitch with it. He thought it was pretty neat. At that point there was sort of an incubation group that he was running with a number of technology projects, and so it continued within that group and Mitch hired several other people. So eventually I sort of handed off the day-to-day development to a team of 3 people who worked off-site. That became *Lotus Manuscript*, which eventually found a home in the engineering and scientific products division [ESPD at Lotus] which they had set up. It did okay, but it wasn't anything like the success of the earlier products.

After that I took some time off. I was still sort of interested in doing a new spreadsheet product. I guess my non-compete had expired, and I talked to a few other people and I got a few of them interested. Actually I went back to Ben Rosen and tried to get them involved. That went a little ways and then didn't really work out. Eventually I ended up going back to Lotus as a consultant working on *1-2-3/G* which is a graphic spreadsheet for OS/2 that they were working on. I consulted with them for a couple of years.

Campbell-Kelly: On a full time basis?

Sachs: Yes.

Campbell-Kelly: And did Lotus *1-2-3/G* become a product?

Sachs: Yes. IBM actually sold quite a few of them to corporate customers. But it didn't ever become an important product because it was targeting OS/2 which was ultimately eclipsed by the much more popular Windows 3.1.

Campbell-Kelly: What did you do after that?

Sachs: That was 1990. I got interested in painting and I taught myself to draw and I started painting watercolors. One of the things I did for painting was to take photographs as a reference. Then I started getting more interested in photography and adjusting photographs on the computer. Then I got more and more worked up in that and lost interest in painting.

I did a prototype which was actually a spreadsheet for images, so each cell would have a little image in it and the formula would say: “Okay, I'm going to do this to the image.” Working with some other people I tried to find a publisher and eventually I got Kodak interested, so I was a consultant at Kodak for a year or two. I actually set up a group here in Boston. We hired some people and then they [Kodak] terminated it. So my contract said if they terminated it, then I get the technology back. They got to keep a copy of it, but never did anything with it. So I sat on it for a while and then I started re-writing it. The Kodak version was on the Mac and I started rewriting it for Windows. In 1993, I started a company called Digital Light & Color. We still sell image-editing software for photographers. [www.dl-c.com]

It was a bit of a humbling experience. I initially kind of hired some people to help sell the product. I was funding it out of my own pocket. It really didn't take off particularly well, and at some point I lost enough money that I decided enough was enough! We didn't really shut it down, but it sort of went to a much lower level of activity. I was getting ready to shut it down and I was approached by a company out in Denver that made printers. They offered to buy it out and they did for a year or two. They eventually decided to sell it back to me.

Then, I was getting ready to shut it down again and a friend of mine approached me and said he thought he could work with me and help me sell it. So we've been doing that ever since - sort of a two person, part time operation. It's actually been making money and doing a little better every year. Not that it makes enough money to make a difference, but it makes enough money to pay for the other toys and trips, and it's something I enjoy doing. I deal with the email and the customer support. A lot of other people are interested in the same thing I am. It gives me an outlet for my creativity. It's sort of striking because I'm not really a spreadsheet user. You write these spreadsheets and millions of people use it, but I don't connect with that. So for me it's a lot more satisfying to write something that thousands of people use, but that I do connect with.

Campbell-Kelly: Do you reflect on why Digital Light & Color is not a killer application and the spreadsheet is? [DL&C is the company name; the program is called Picture Window]

Sachs: Well, *Photoshop* is a killer application. Adobe was there first with much better marketing and they put a lot of money behind it. They targeted the printing and publishing industry as opposed to photographers.

But a lot of times it's who gets there first and becomes the *de facto* standard. Certainly Lotus was a huge beneficiary of that phenomenon. But I was never really willing to put anything like the amount of money into it that Adobe put behind *Photoshop*. And *Photoshop* is probably a lot better product for a lot of people. We targeted a very specific segment - serious amateur and professional photographers. We have a fairly happy user base, but it would be very hard to grow into anything like the *Photoshop* base.

Campbell-Kelly: One last question. People try to improve on the spreadsheets -- and Lotus *Improv* was such a product -- but none of them seem to succeed. Why do you think that is?

Sachs: Well, it became a mature technology, like the word processor. I mean, people don't try to make a super-duper word processor any more because word processing basically does such a large percentage of what people want, that it's enough. When you keep adding to it, you actually make it worse. Again, the industry is so totally different than it was back in the 1980's.

I mean, innovation was a big deal back then. Now innovation is really a dirty word. If you have an organization with 50,000 copies of some software, the last thing you want to hear about is a new release, let alone a new product -- because the upgrade costs and training costs are phenomenal. In the old days, it was like one person was buying one package, and if they got bored with it, they wanted something new. But we're in an entirely different world where IT -- a department that used to control everything in the mainframe world and then totally lost control for a few years -- have heavily reasserted themselves and are now kind of back in control. They dictate what's on a PC and the company policies and so on. So it's a very different world now.

Campbell-Kelly: Thanks so much, Jon.

Sachs: You’re welcome.