Oral History of Norman Meyrowitz

Interviewed by:
Marc Weber

Recorded: July 29, 2008
San Francisco, California

CHM Reference number: X4968.2009

Marc Weber: Today's the 28th, is that right?

Norman Meyrowitz: Twenty-ninth, I think.

Weber: Twenty-ninth of July 2008. I'm here with Norm Meyrowitz. If you could just really start at the beginning. A little bit about your background; where you grew up and what got you interested in computers.

Meyrowitz: I was born in Brooklyn, New York and until the age of seven lived there. Then I moved to Wantagh, Long Island where I concentrated a lot in high school journalism. I won some awards in high school journalism both for writing, and also for layout and design. That kind of sets a stage. I went to Brown University and freshman week, I went to a seminar with Andy van Dam who was the Chairman of Computer Science. I think I was the only person who turned up for the seminar. So we were just talking, and I talked to Andy about newspaper layout and design, and asked if there were any computer programs. I had actually done some computer programming in junior high school and high school on an old timeshare DEC-10 with a Western Electric teletype and paper tape. Anyway, I asked Andy about newspaper layout and design. We talked about an Atex system the newspapers were using, but Andy told me later that he knew I was the guy who he was going to corral into doing the next generation of text processors in hypertext. It turns out that Brown has a long history of hypertext. Andy got there in '65 or '66 and very soon he met Ted Nelson. Ted and Andy both went to Swarthmore [College] and they knew one another through that connection. Ted Nelson is the guy who coined the word 'hypertext' for non-sequential writing. Ted was wild. If you think of Douglas Adams and *Hitchhiker's Guide to the Galaxy*, he was several orders of magnitude crazier in terms of just his being. But he was incredibly brilliant, and incredibly creative. He wrote a book called *Computer Lib/Dream Machines*, one side—the other side he talked all about hypertext and computer graphics. This was in the mid '60s. Actually, *Computer Lib* might have been a little bit after '70. In '66 and '67 Andy and Ted got together and built a hypertext system on the mainframe computer at Brown which was a very, very old computer. A couple of years later, Andy started a new project called FRESS, File Retrieval and Editing System. That turned out to be a tour de force in '67. It ran a CP-67 timeshare computer and it used a light pen terminal that had to be actually booted up with a paper tape. It displayed multiple columns of information and you could just touch on a light pen and follow a link. It looked very much like the web does, but without color and with calligraphic characters because it was a vector calligraphic display. It actually used beams of light to draw the characters. There were a couple of courses taught on using FRESS. One was on English poetry, and the kids would go to class, and then they'd go to the lab with this machine, and they'd read some poetry and they'd actually read comments that other kids had written. You could just point it at a piece of text and write your own comment and that link would be there for everybody to see. There was an entire corpus of information that everybody could interact with. It built this whole interesting web of comments.

Weber: Did you ever see FRESS? Was it still…?

Meyrowitz: I was the last person to actually give demos on FRESS, and—

Weber: Wow!

Meyrowitz: There's a funny story. I would give this demo, and by that point the system was a little bit fragile because Brown had changed the mainframes. I was giving the demos up until '77 through '81 when I went to Brown. I would give this demo and it was always the same demo. John Donne or Bryon, I think, was the poet. I'd always follow the same link from the student name KMA. There were always interesting comments and I'd just go zooming around the corpus and everybody was duly impressed. In 1983, I got married to Katherine Marie Archambault and some day like a couple of months after we got married we were just talking about hypertext and suddenly it dawned on me that I had actually been following her links in giving this demo. She had taken this poetry course and just through coincidence the demo I had been giving was following all her links. So it's just a fun story to know about. We had actually met through hypertext before we even knew each other.

Weber: Was she interested in it or it was just a course she took?

Meyrowitz: She wasn't particularly interested in it. She was just taking the course and it had this feature and she went along with it. She thought it was mildly interesting except the system didn't stay up all that much back then, so it was a little annoying. She thought it was pretty interesting. The funny or ironic thing is, in '83 and '84 and through the '80s, I started developing a system called Intermedia, which was a much more sophisticated hypertext system. It was ten years later. Kate thought, "You're doing hypertext again; wasn't that already done?" At Brown, hypertext had been going on since the mid '60s, and so for me it was just a given that this was how things were going to work. There was going to be a worldwide corpus of information that everybody could link to, and it was really just a matter of time. So Intermedia, which was mid '80s to the end of the '80s, was right after the Macintosh had come out and [the] desktop metaphor of computing was really taking hold.

Weber: Sorry, I need to back up a little bit. At Brown you were working with Andy and what was your work on?

Meyrowitz: At Brown I was working with Andy, and as a student I created the first Unix-based window manger called BRUWIN [Brown University Window Manager] on a Ramtek 64K display and it was fun. It used a puck on the digitizing pad which is the forerunner of the mouse that everybody used to do and it was a beautiful color display. I think it was 1024 by 768 at that point; the Ramtek display. It was connected to a Unix-based VAX-11/780 as I remember. It was a window manager. You could run virtual terminals inside of a window. That was one of the things I did. I demoed FRESS and was a Research Scientist in the Computer Science Department. For the first few years, it was looking at object oriented programming and Smalltalk and more window manager stuff. In '83 we did a bunch of fundraising and got a lot of money—about $50 million—from IBM to do campus-wide computing. Brown had one of the first broadband networks back in 1981 or '82 called BRUnet. We wired the entire campus with coaxial cable, the same as cable is used today. There were these modems from—boy, I can't remember the name of the company—but they were just like the modems that people use today. You would attach them to the cable and then you would get an Ethernet—not an Internet because there was no Internet yet in terms of pubic use. You get an Ethernet connection [and] the entire campus could be connected. Largely, it was used for terminals. It was pretty impressive. Anyway, we started a research institute in 1983. I was one of the co-founders. It was called IRIS, the Institute for Research and Information and Scholarship. Besides the whole campus computing push—that was part of what IRIS did—another thing we decided was to create the next generation hypertext system called Intermedia. We just thought this was inevitable and somebody needed to build it. So the desktop metaphor, that Apple introduced with the Macintosh introduced in a wide

scale after the Xerox PARC experiments, had taken off. We decided that the way hypertext should really work was that hypertext should just be built into the operating systems and that every application should have the ability to be the target or the source of links. You make a selection in one document and you'd say—just as you say, "Copy" today—you'd say, "Create Link." Then, you'd go to another document and you'd say, "Finish Link," and you'd actually have these two things linked together. We did that. We had a text processor. We had a graphics editor. We had a timeline editor to do historical timelines. We had an animation editor to do some vector-based animations, and all these things could be linked together. Eventually, the system was used to teach a biology course and an English course. This time, the system was not on a mainframe but was on workstations, and those workstations were linked together on a local network using NFS, Network File System. Everybody could read the same corpus of biology or English information. They could create new word processing documents or graphics documents and link them into the system, and everybody on the network would see those. It felt a lot like the web before the web.

Weber: You could link, because you had talked about being able to link to a document in any program. How, for instance, would you link to a word processor document to put the anchor inside of it?

Meyrowitz: If you wanted to link you would select something—it could be a character, a word or an entire paragraph—and you'd say, "Start Link." Then, you'd just browse around the way that you do after you do a copy today in Mac or Windows. When you found the target of the link—let's say it was a graphics object, a rectangle or a couple of rectangles—you'd select them and say, "Complete Link," and now those were linked. Even if you edited the word processing document, the anchor, which was what you had selected, would stay in sync. If you added a paragraph, the anchor would still stay in sync, and if you moved the rectangles around they would stay in sync.

Weber: But the documents had to be created by Intermedia?

Meyrowitz: These documents had to be created by Intermedia. We were doing some research with Apple and Claris, and we said, “What you really need to do is build this into the Macintosh operating system so that all the applications could take advantage of this,” and Apple would help fund some of Intermedia. We really pushed and pushed. They were thinking of doing it and then they decided to do Publish and Subscribe which was the big thing in the late '80s. Publish and Subscribe is where you could copy a chunk of a document and put it into another document, and if the first document was updated the second one would be updated. It turns out Microsoft spent an enormous time doing that, Apple spent an enormous time doing that and very few people in the entire universe ever used it. It was kind of a bummer that they went Publish and Subscribe rather than hypertext. But this industry is more about—a lot of it is luck, including much of the stuff that I have done. Luck and faith; you make one decision and you really don't know if it's the right decision until several years down the road.

Weber: Did you have any dealings with Bill Atkinson when he was doing HyperCard?

Meyrowitz: I met Bill Atkinson before the Mac even came out, and when he was doing MacPaint and they were doing the operating system. One of the ways the Mac got started was that Steve Jobs and Dan'l Lewin had decided that they would create a university consortium and there would be 24 universities in that consortium, each that pledged to do two million dollars of sale of computers over the course of a couple years. Before the Apple Macintosh launched in 1983, they went around and showed the Macintosh

to everyone in these universities. Twenty-four of them signed up. Apple, I think, had $52 million of sales before the Macintosh started. We had all met Bill Atkinson doing MacPaint and other folks before that happened. The HyperCard stuff snuck up on us. Even though we were doing research with Apple and we were trying to get them to put hypertext in the operating system, maybe a week before it was announced we learned about HyperCard. HyperCard was a whole different kind of kettle of fish. Instead of it linking disparate documents from different applications it was a whole embedded application where you had stacks of cards and a scripting language that would allow you to not only link from card to card, but also to do problematic scripting so when you hit a link or hit a button on a card various different things could happen. It wasn't really as much of a hypertext system as a programming language where moving from card to card with scripting was the paradigm. It was more like Javascript in a bunch of stacked web pages. That was kind of a predecessor to the web in some sense. There was also another application that was called TEAOMOS [Telio?] [ph?]. I can't remember the name—File or something or other—that was an application on the Mac before HyperCard and HyperCard borrowed a lot from that. That one was based on a database, but from the database you could populate various different screens or cards and have full applications. It was very, very cool back at that time even before HyperCard.

Weber: Did HyperCard divert Apple away from real hypertext, do you think?

Meyrowitz: It diverted them away from real hypertext and doing stuff over the network, because it turned out that HyperCard was this insular environment where you exchange stacks of cards. At that point, they had the AppleTalk network and they could have started doing stuff over the network. Eventually, they canceled HyperCard because it wasn't really generating any revenue, even though it was kind of interesting. I wouldn't fault Apple [for] going the HyperCard route. I think they were equally interesting. It's kind of, “Should I have a dog or should I have a cat?” You can have both if you want.

Weber: In Intermedia you could connect to remote servers so it was fully—

Meyrowitz: You connected to—

Weber: I mean over the LAN [Local Area Network].

Meyrowitz: —network file systems over the LAN, so essentially the file system—this is how the SUN NFS worked: you mounted the file system on all the different computers as if it were a hard disk, but it was actually over the network. You actually had everything mounted and it was a different paradigm than what the web eventually came to be, where there was a protocol between machines that passed bits of information back. Every computer on our network was actually reading a shared file system and writing that same shared file system, and you had appropriate locks so that two people couldn't write the same file at the same time. All the links were kept in a database. It was originally an Ingres database and then it was some type of B-Tree database. We also had the locks there so as people made links there was consistency.

Weber: No dangling links. No broken—

Meyrowitz: Yes, no dangling links. You couldn't wreck the database. Another thing that Intermedia had, we had full text indexing. We indexed the entire corpus of information, all the text that was in everything. It's really interesting. That was in the late '80s and now it's 2008, so it's 20 years later. The thing that we were worried about at that point was how much disk space you could take. We didn't have very much disk space and you couldn't afford to have that much of the disk taken up with a inverted index, which is how the indexes are used today. There was this algorithm that we got from a paper where you would take all the words, hash them eight times and then do a bit compression of all the hashes. If you wanted to search stuff, you would figure out what you would pass, what you wanted to search, search through all of those little hashes and you'd get a bunch of hits. Now what it meant is not all those were hits, but the ones that didn't get hit definitely weren't and then you'd have to go and search through all of those with text to actually find out if that word was there or not. But it was quite, quite cool. We got instantaneous response, and we used all this morphological analysis so when you type stuff in you could actually deal with plurals and different tenses and parts of grammar. [It was] Google 20 years [ago]. Probably ten years later everything had change drastically. CPUs… are—there's Moore's Law—probably a hundred times faster, disk space there was a hundred [times] more. I think our disks started at like 10 meg[abytes] and they might have been 25 meg towards the end as opposed to half a, gigabyte or three-four gigabytes. So one of the things that Intermedia had, which was very interesting, is it had back links. When you created a link between a text processing document, and another text processing document, you could follow the link from number one to number two. Then you could also—if you were in number two—you could follow the link to number one. They were bidirectional, and the web doesn't have that. You follow a link to one document and that doesn't mean you can go back from the second document to the first document. We said we had to have a database and later on, we never thought there'd be the computing power that you could search the entire corpus of information even on a single computer and compute the back links. That's how Google works. If you want to find back links, they've gone through a hundred billion or 50 billion documents and they've actually computed where all the links go and where all the back links are. We never thought in a million years you could deal with those back links. One of the interesting things was in '87 or '89, I had given a keynote—I think, in '87—for the hypertext conference and it was called, *Hypertext-Does It Reduce Cholesterol, Too?*, because all of a sudden that became the big word. I still thought it wasn't really hype, it was really important. At that point—maybe it was '89—I had a list of the things that had to happen; ten things that I wished for. One of them was that Al Gore was sponsoring a bill for a network to do supercomputing, and I said, "Boy, wouldn't it be great if part of that was used for hypertext?" I don't think I am particularly brilliant, it was just what I thought was the natural progression of things. A couple of years later I gave an advanced tutorial on hypertext link systems. It was in Paris, as I remember. I was talking about our database with forward links, back links, keeping everything consistent using a database, and using a shared file system. There was this guy in the back who just kept asking a zillion questions and arguing over whether that file system and database thing was good or whether a system that had a simpler protocol where the links were embedded in the documents was better. They were one way, there was no database, you could create a link even if the target document or the computer was down; it didn't require any communications. It was all just staples. The way we did [it] here you didn't have to have any other staple but what was on your computer and then eventually you'd connect to another network. I eventually found out that that was Tim Berners-Lee, and he turned out to be right. Our system was a little bit too complicated. It was still thinking in the local area network point of view because that's all there was in the early 1980s. There was still a smidgen of the Internet. Tim was at CERN and they had access to the pre-Internet. They had this vision of things connected. I guess I had used FTP and others, but there wasn't a real connection that an FTP like system, or a Gopher like system, which were also the predecessors of the HTTP protocol. It seemed like it was a hokey way to get documents from here to there instead of using a file system. Again, we never expected 100 megabyte networks in people's homes and backbones that were ten gigabit networks that were prevalent across the entire country. It turns out once you do that you want to have a simple protocol and so we had the right user interface but the wrong architecture for 20 years later.

Weber: The HES [Hypertext Editing System] and FRESS also were single machines with...

Meyrowitz: HES and FRESS were all single mainframes.

Weber: But with multiple terminals theoretically more than one person could edit the same document?

Meyrowitz: I don't recall. I think you could only do one at a time.

Weber: HES was still running when you were...

Meyrowitz: I never saw HES run. HES is Hypertext Editing System, which is the one that Andy did with Ted Nelson.

Weber: Incidentally, I have the FRESS manual that Andy gave me a copy.

Meyrowitz: I have some FRESS transcripts and somebody wrote a FRESS emulator that still may exist. Steve DeRose wrote it and it still may exist. I actually had a transcript from a session that I did on FRESS probably five, ten years ago using that emulator. I have that somewhere that I can give you.

Weber: That's all of extreme interest, yes.

Meyrowitz: So.

Weber: Ted Nelson's vision originally had been, basically[?] hypertext but that was very far out at the time.

Meyrowitz: Yes.

Weber: I take it [that for] Andy and his students, including you, this was something that was kind of pie in the sky from Ted Nelson. You weren't really trying to implement the—

Meyrowitz: Ted Nelson really is a visionary, and visionaries often don't have a great sense of time or how long it takes to do something or what the prerequisites are. Plus, Ted is kind of a jokester. He's a little like Hunter [S.] Thompson; he's that kind of character. The Xanadu Project, which was his hypertext system—I think that the name is just a joke. Xanadu was the poem that never got finished and for Ted, Xanadu was something that was never going to get finished. It was just a hobby or a vocation but he never meant it to actually ever get completely finished. We also knew about Doug Engelbart who had created the mouse and then later created a system called—

Weber: NLS [oN-Line System].

Meyrowitz: —NLS, and it was a beautiful system in its time. I may have an NLS manual. Doug's system was really an outliner that ran on a bunch of different computers. You could create links between documents and they even had video conferencing in there. They did this incredible experiment in '68 where he was down in Menlo Park at SRI and one of his folks was up in San Francisco, and they video conferenced one another. This was in 1968. We knew about that type of networking and that type of distance connections. He was using incredibly expensive, very hard to obtain networks and hardware at that point.

Weber: Andy van Dam was not holding this out as a design goal. This was something for the distant future.

Meyrowitz: We knew it was going to happen. Brown was in the forefront of networks, as I said with BRUNET, and in the forefront of workstations. I believe we had one of the first, if not the first, Apollo system, which was a system very much like Sun workstations. They were big competitors in the mid '80s, I think.

Weber: Didn't Intermedia run on Apollo?

Meyrowitz: No, Intermedia didn't run on Apollos. It ran on, of all things, Sun and IBM workstations using a version of the Macintosh APIs that ran on Unix, and—

Weber: Oh, the Apple Unix?

Meyrowitz: It was before Apple Unix. There was a workstation company called Cadmus that was producing workstations largely for CAD and Tom Stambaugh took the Macintosh APIs and reprogrammed them to run on Unix workstations. We went up to Cadmus and convinced them to license that to us for the university. We were using Mac APIs on Unix systems because we wanted systems that supported the network file system, that had virtual memory, that had bigger disks. So we did that and it was a beautiful system. Eventually, convinced Apple to create a A/UX [Apple Unix] which was a Unix system and we ran on that. Cadmus went out of business, and we had the only license to CadMac, and Apple wasn't that happy to have it around. We had actually convinced them that they should release it to universities, because having that API around, especially if they could sell it to other companies to put it on their workstations, could actually be lucrative. We almost convinced them to do that. We were actually having a celebratory dinner at McArthur Park in Palo Alto to celebrate this and then Jean-Louis Gassée walked in and said, "The deal is off." That was before Microsoft was anyone.

Weber: What year?

Meyrowitz: Oh...

Weber: Roughly.

Meyrowitz: Gassée was there so it must have been '86 to '88. It was before Windows took off, because Windows 3.1 didn't come out until '90 or something. It's all quite interesting and a lot of the history of the Internet and hypertext was just fate or a decision that didn't seem that drastic at the time.

Weber: Just to finish up with hypertext, you went to the hypertext conferences that Wendy Hall helped organized, right?

Meyrowitz: We started the hypertext conference and Wendy Hall was one of them. I was one of the founders and Frank Halasz and Randy Trigg who did NoteCards at Xerox PARC. And there was—

Weber: Herman Mauer?

Meyrowitz: I don't—

Weber: <*inaudible*>

Meyrowitz: There was a guy who did something that looked like HyperCard in Pittsburg. There was an educational software sort of HyperCard-ish system about ten years earlier that ran at University of Illinois at Champaign.

Weber: Oh yes?

Meyrowitz: I can't remember what it was. It had some Greek God's name, as I remember, and I can't—I think it starts with a C, but I can't really recall right now.

Weber: But you knew all these people.

Meyrowitz: Oh, we knew all these people and the conference turned out to be— I think the first one was in '87, and I gave the keynote in '89, and by that point there were a thousand or so people coming to it, because hypertext was big. There was a system that came out called AVIL [ph?] which was done by Peter—don't remember his last name— they came out of England and ran on Macintoshes. It was single user. It was to create a corpus of information that was, linked internally, not networked but it was quite a nice system with an outliner. Around that time some of the outliners came out. Doug Engelbart's system had an outliner and links. Then outliners dropped off and it was just text processing type things and links. Then, David Winer came out with his outliner that ran on the Mac. I can't remember what the name of that was, but that was quite interesting. Dave Winer turned out to end up being the guy who coined the word "blog". The community is kind of quite eclectic; they can do more than one thing. In terms of hypertext, that was the history up until I left Brown. Intermedia turned out to be widely known. We published a lot of papers. It was one of the first things that used object oriented programming, so it also had that whole side of it. It not only was hypertext, but it was object oriented. It built these building blocks that had a text building block and a graphics building block that you could actually build other applications out of, which now is the norm, but at that point it wasn't. At some point there was a recession in the early '90s and

funding dried up for universities. IBM and Apple and others had been giving a lot of money. At some point I decided, "It's time to try industry." I knew some of the people at GO Corporation. I knew Dan'l Lewin who was the VP of marketing, at that point. He had been the guy who had executed the Apple University Consortium for the Macs. He had actually been with Steve Jobs at Next. Dan'l convinced me to go out to GO. Bill Campbell, who I knew from Apple, he was a VP of marketing at Apple.

Weber: Claris.

Meyrowitz: Then at Claris after that. He became the CEO. I knew of Jerry Kaplan who had done Agenda, which was the first really cool PIM, Personal Information Manager, in that you type stuff in for calendar events and database records, but you could do all sorts of searches on them. You could program searches so it was very natural. Robert Carr was there. He was the VP of engineering. Jerry was the founder. Robert Carr had done a system called Framework which was one of the first window manager type systems and it ran on PCs running DOS. There are all these very cool people there. I talked to them and decided I would go out there. Microsoft had been talking to me, but the GO stuff just seemed out of this world. The whole idea was you would have a tablet computer with no keyboard, and you'd have a pen, and the system would do handwriting recognition and so you could just write on this. Lo and behold, you'd have a computer that ran totally on your handwriting and pointing and clicking. This is a second generation. GO had done their own hardware and then when Bill Campbell came along they decided they would get out of the hardware business and go into the operating system business, which had started becoming lucrative as Microsoft came up. The company split; EO was the name of the company that split off. This is from 1992—this computer. *<holds up computer>* It looks just like the tablet computers of 2008.

Weber: You joined around '92 then, right?

Meyrowitz: I joined in the middle of 1991, so they had been going on for about three and a half years, but had just switched to an operating system company when I got there.

Weber: You just hold it steady. Yes.

Meyrowitz: Is this good?

Weber: I wouldn't call it legible, but you can certainly see that it's got text on the screen.

Meyrowitz: Oh, I meant the interview—

Weber: Yes, very good. I don't think the screen display is good enough to really demo.

Meyrowitz: Oh, I'll see if I can change the contrast. Can you see any of it?

Weber: Okay, let me zoom in again. Yes, actually that's legible as long as you don't change the angle much, but that's quite legible.

Meyrowitz: It had a notebook metaphor and it turned out there were hypertext and links even embedded in those applications where you could actually create a link from one document to another. Very few people know that, but that was something that was in there before I got there. I largely ran all the OS that was non-kernel for the last year and a half and mostly spent time getting the product, the operating system, out the door. At that point, besides th the EO—I don't have that attachment, but there was an attachment on one of the EOs. I don't know if it was this one or the bigger one, but it had a cellular phone attached to it, so it really was a mobile computer. It had a cellular telephone. They were all analog at the time. It was one of those big phones. It was a full sized headset. The EO had a little cradle and you could carry it around. It was probably 50 times bigger than today's cell phones, and since it was analog it essentially simulated a modem and so you ran at modem speeds. At that point modem speeds were 14.4 or 33.6 [kbps]. It was still pretty slow.

Weber: Ninety-six-hundred.

Meyrowitz: Maybe even 9600 baud, but it was really cool. We had applications that were spreadsheets that were these very cool PIMs. There were graphics programs. There were people using it for insurance claims. You could use a modem to actually send files back and forth. There was an email program to send email back and forth. It felt just like, say, a Palm system does today. The only problem was that battery life wasn't very good, the CPUs weren't very fast, the systems were heavy. But it was a beautiful operating system. It still works today; 15 years, 16 years later, this machine is still working. The mobility was Jerry's idea. He was sitting on a plane using Agenda and having to type and the seat was tight and the laptops were big at that point. I don't remember if he even had a laptop, but he was saying, "Wouldn't it be nice if I just had something like a pad that I could write on and have my pen. That was the first notion of GO and the PenPoint operating system.

Weber: He wasn't aware of the Dynabook type stuff.

Meyrowitz: I think everybody was aware of Dynabook; Alan Kay's notion of a laptop with a keyboard, which he probably did in the mid to late '60s. I think he was at the University of Utah, at that point. That was the vision that everybody wanted. There was another hypertext system that I didn't work on. I was working on the window managers. It was called the—

Weber: Take that down.

Meyrowitz: <*puts down computer*> I can't remember what it was called; the electronic document system.

Weber: The one for the Navy?

Meyrowitz: Yes, it was done for The Office of Naval Research. It was to show what repair manuals could look like on the USS Carl Vinson, which was an aircraft carrier. That was another hypertext system. It was much more graphical, also done on the Ramtek. AI can't remember my train of thought—

Weber: We were talking about the Dynabook.

Meyrowitz: Oh, you were talking about the Dynabook. Before that system got created, Andy had an industrial design prototype of what the system would look like and it was a laptop. It was a plastic model of a laptop with a 1024 by 768 [pixel] screen, a keyboard. It had hinges so you could fold it. This was in '83, '84, so all of this stuff was pretty evident to the people who were working in both hypertext and kind of in mobile computing. Before laptops were a big new thing, everybody thought they would come. I'm sure Jerry Kaplan knew about the Dynabook, but he wanted to go further than the Dynabook. He didn't want to have a keyboard.

Weber: But you think the pen was not a common idea before he came up with it?

Meyrowitz: Not really. I believe that the light pen had been a common idea in the early '60s, but that required you to hold your hand up and touch a screen and usually the screens weighed 150 pounds so it wasn't anything portable. The pen was the point of the GO operating system. That was the slogan, "The Pen's the Point.". I'm sure people had thought about it, but no one had really thought about implementing it, because it was very high-tech back then. Digitizers just came to the floor. There were some great engineering folks on it; Celeste Baranski was the engineering manager on the first GO machines and on the EO machines, and she went on to be at Palm, at Handspring and then at Palm again and doing the Treo and all the Handspring stuff that went before. She was a great hardware designer. It was Mike Ohway [ph?] who did hardware design and software. There's Kevin Doren who was actually the VP of engineering who lead that whole hardware effort to begin with. He went on later to manage some of Paul Allen's—the other co-founder of Microsoft—venture companies. Jerry knew about this stuff and the innovation was, can you have something mobile, can you have handwriting recognition, can you have cellular phones so there could actually be communications, and all that came to the floor. Can you actually get this operating system to run on a lot of disparate hardware from different manufacturers? At that point Grid was making a system; IBM was making a system called the ThinkPad. The ThinkPad—which became IBM's brand name for all their laptops—was actually the name of their tablet. Eventually, they stopped doing their tablet and used the name for the laptop. It ran on the ThinkPad, and ran on Grid, it ran on Toshiba tablets, it an on Hitachi tablets, and all we had to do was create a BIOS [Basic Input Output System] to run on each machine for Windows or DOS, you have to create a MIL, called Machine Interface Layer, for each of these computers. All those companies created the MILs and we had this running on all these systems. We even had a Japanese version running. One thing that no one at GO realized—nobody could actually know that we were trying to sell the operating system to all these companies. I think it was $35 a crack. That was how we were going to make our money. We weren't selling hardware any more, but eventually we found out. I guess companies eventually told us. They probably weren't supposed to, but they told us that they'd love to buy our operating system, but Microsoft had a system called PenWindows. They just whipped this up really quickly and a guy named Jeff Raikes, who went on to be president of the entire Office division of Microsoft, was the guy who came to GO and he said, "Bag your operating system. Use ours." It was mostly smoke and mirrors, but two things. One, Microsoft was pushing PenWindows—this was back in '92—on all these hardware manufacturers and so they were getting a lot of pressure. Microsoft was supplying their regular operating systems. Second, the thing that we didn't realize is that all

these manufacturers paid Microsoft a royalty on every machine they sold even if it didn't have their operating system. To put PenPoint on one of these tablets there would be a double tax. You would have to Microsoft even though you weren't using any of the Microsoft stuff and you would have to pay GO. So GO died not because the technology wasn't good, but because Microsoft cut off its air supply, which was what they were doing back then. None of these manufacturers could afford to pay twice so they'd go with PenWindows. As soon as we died, Microsoft forgot about PenWindows and only like 15 years later, Microsoft had this brilliant idea of tablet computing. I thought it was hilarious that all of sudden 15 years later there was this new innovation that had just sprouted in everybody's minds. The other reason that tablet computing didn't really go very well back in the early '90s is battery life wasn't good.

Weber: How long?

Meyrowitz: Maybe an hour and a half. The big innovation that has never happened in computing isn't CPUs, they always get faster, isn't networks, they always get faster. It's battery life. That really hasn't improved very much. Also screens haven't really improved very much. The resolutions of screens are still not good enough that things can look like magazines. The resolution of this was black and white. The resolutions weren't particularly great. The handwriting recognition wasn't great. The handwriting recognition would be like '94 percent accuracy, which sounds great, but it means that in every hundred characters there are six errors. If a word is five characters long it means that in every 20 words there are six errors, which is actually lousy. Plus it took a lot longer to do that than to type. It wasn't clear that handwriting recognition was going to take off. We had both character recognition and later on script recognition.

Weber: Cursive.

Meyrowitz: Cursive, but it still wasn't good enough. At this point Jeff Hawkins was at Grid, and they had their own operating system. We put the PenPoint operating system on the Grid computers. Jeff and those guys, they were creating some applications for PenPoint, as I recall. Then, GO couldn't sustain its revenue and ran out of money. AT&T bought EO, who was producing the hardware. EO acquired GO. Another half a year went buy and the plug got pulled. At that same time, Jeff Hawkins and Donna Dubinsky and then Ed Colligan—but first it was Jeff and Donna, because they had talked to us and Joel Jewitt.

Weber: It was Jeff and—? Donna joined soon, but...

Meyrowitz: I remember, at some point when they were still doing applications for PenPoint, Jeff and Donna visited and soon after that Jeff had a epiphany. PenPoint was this huge operating system that had virtual memory tasks, all sorts of APIs—that's the EO that is shutting off [the computer]—power management 15 years early. You didn't need a full multitasking virtual memory operating system with a book this thick of APIs.

Weber: But the EO and the GO was intended to be a replacement for a PC not an accessory, right?

Meyrowitz: It was intended to be a replacement for the PC, but Jeff thought that it's not going to be a replacement for the PC for a while, because battery life isn't there, the handwriting recognition isn't there, no one's going to buy PenPoint because Microsoft is going to kill it one way or the other. You could take some of this technology—the handwriting recognition—and do two things; create a smaller device that was used for maintaining your daily life, for contacts, calendars, what have you. Then he came up with the idea. It was a very small operating system, a small number of APIs, no multitasking, no fancy stuff. The other thing he decided is the reason handwriting recognition is hard is that a lot of the characters in the English character system look like one another and so it's very hard to distinguish them. He said, “Instead of trying to distinguish these”—which is an impossible task, and now in 2008 it still is very hard. It still is not perfect at all. —“we'll teach everybody how to write differently." So he invented Graffiti, which makes sure that every character that you wrote was easily distinguishable from every other character. That really set off the next generation of mobile computing because you could keep these things in your pocket, and you could write on them, and hold them in your hand, maintain your calendar. The third thing that he did was a notion of sync. Now, we had all been—back in the PenPoint days—creating calendar formats so you could exchange calendars and appointments with one another just like the .vcs protocol now where people can click on something, open it up, and the calendar entry gets put into Outlook. There were those protocols there, but what Jeff decided was if we just had a sync protocol where you plugged your Palm handheld into your regular computer, and just hit a button and said “Synchronize,” now all of a sudden stuff was synchronized. There was a set of Palm applications on your desktop that were synchronized, and then you had the stuff on your handheld. It was beautiful. It was a seamless system. You could create stuff on your desktop, plug in your Palm, it would move over to your handheld. Create stuff on your handheld, plug a cable in, press a button, it would sync with your regular desktop or laptop computer. It was those innovations, the sync, the handwriting recognition, that people had to re-learn but it was much better at discerning your handwriting, and the notion that it was a very little operating system so you could kind of produce it cheaply. You didn't have to spend enormous time with an enormous software staff creating all this stuff, and you had applications that people really needed. That was kind of the next generation of mobile computing.

Weber: Were you aware of this? When did you become aware of this?

Meyrowitz: I became aware of it six months before they launched because there are no secrets in Silicon Valley. That was really interesting. In mid '93, I joined Macromedia, first as Director of Advanced Technology and then very soon I took over a bunch of projects; Action, Sound Edit and then very soon Director. Director is kind of another take on both the hypertext and the HyperCard.

Weber: At that let me change the tape.

Meyrowitz: Okay.

Weber: It's a good point to change it, because we're at zero minutes.

Meyrowitz: Are you getting what—

<audio ends abruptly>

END OF TAPE 1

START OF TAPE 2

Weber: You're talking about, oh, one last Pen Computing question.

Meyrowitz: Uh-huh.

Weber: You didn't mention Newton at all. How important is that? Apple kind of pulled a fast one, right?

Meyrowitz: No, it wasn't necessarily a fast one. At that point, Bill Campbell had come from Apple and then later on he hired Mike Homer, who was the VP of marketing. Mike, who was brilliant, one of the best marketeers I ever saw. He was eventually at Netscape and also driven and thought that everything could be done in a second. He knew a little bit about Newton. In mid '92—right after we had shipped this multitasking industrial strength operating system for Pen Computing that IBM would sell and use for insurance adjusters and medical records—he said, "We have to build something handheld that regular people would use for their calendars and stuff." It was completely orthogonal to what we had done before and very, very difficult because these would be small systems. You couldn't get virtual memory. You didn't have a disk, etcetera, etcetera. That was a project called Amstel [ph], and we undertook it essentially because we had heard about Newton through other people. When Newton came out, we were towards the end of our tenure, and Bill was towards the end of his rope. I thought that Newton was interesting but again it was, it had some of the same handwriting recognition problems. Steve Capps did a large part of that operating system. There's a lot of operating system; fancy stuff and different objects. I can't remember it, but they had their own language. There was a lot of concentration on the architecture and the software, and not as much concentration on the applications and what you did with them. It was a minor success for about a year or two and then it got killed off. I can't remember when it got killed off. It was before Jobs got there. Then that group had moved on to creating these clamshell systems for education as part of the mobile education [segment]. It was K through 12 clamshells that I think they shipped. Then Steve came in and just pretty much killed everything except Macintoshes for professionals, laptops for professionals, Macintoshes for individuals, laptops for individuals. That was always what he said he was going to do, and ten years later that's what he's doing.

Weber: So you jumped to Macromedia before?

Meyrowitz: I jumped to Macromedia in March of '93, as I recall—maybe May of '93—and started working on managing the Director Project, Director for Windows, then future releases. Director had color. It had animation. It had started in the mid 80s by Marc Cantor, who was another brilliant character. He had a vision on the original Macintoshes before they were color, of having bitmap animation where just a clipbook of animation that a kid does looking through a book, you can actually have animation by changing pixels. He had some incredible effects for all these fades and wipes and barn door effects, which were like movies. He did that and then it went to color. There was Jay F—I can't remember his last name—who worked on it. There was JT, John Thompson, who worked on it for a long time, who in fact invented the language Lingo. Dave Kaiser, was the VP of engineering for a while doing early versions of that; and then Joe Dunn was the VP of engineering, who hired me. Then I became the VP of engineering. Director, by

the time I got there in ’93, was in full color and we were trying to get it to run on Windows 3.1, which was the really big shift when Windows became industrial strength. That was a hypertext system, but it was more like the HyperCard where you had a bunch of screens and you had a programming language that allowed you to create buttons that allowed you to link to other screens.

Weber: A lot of multimedia programs in the late ’80s did have some sort of linking mostly from . . . didn’t Director always have some sort of links?

Meyrowitz: It had more buttons that could take you from screen to screen. So it had the link; I was just talking about Director from the early days. From its second release, it had primitive buttons. First, it was just animation and then it had primitive buttons.

Weber: Tell me if I am right to think that the early examples of hypertext were obviously the systems designed for online help. Some systems had hypertext <*inaudible*> and multimedia and that was about it, correct?

Meyrowitz: Although there were self-contained things like OWL, you know...

Weber: Right.

Meyrowitz: Those hypertext systems, there was another one by Mark Bernstein, I believe, that was a hypertext system.

Weber: I mean there were many dedicated systems, that had a common use for people with CD-ROMs running multimedia they were exposed to the concept of the link—

Meyrowitz: Right, but the CD-ROMs didn’t come to be until the early ’90s. CD-ROMs existed, but they really weren’t that prevalent until around ’91. I recall right when I got to Macromedia was when Apple announced that they were going to include CD-ROMs drives in all their computers. That’s when things took off. When I got there the previous released Director 4 had been the first that ran on CD-ROM and then we were actually really gearing it up to run well on CD-ROM on Windows, as well as on Macintosh. It had only worked on Macintosh up until ’91. That turned out between ’91 when that shipped and say ’95, when the Web came to be, that was the in thing to create CD-ROMs. They all had links and there was this great thing by Joe Sparks called Total Distortion, which was a CD where you, essentially, zoomed around virtual worlds, played games in the virtual world. It looked a lot like the video games now, but it was done on a CD-ROM in ’92, ’93. There were zillions of education CD-ROMs and games CD-ROMs all done, Director, and it was the rage. They looked beautiful. They even had video ’cause there was AVI, the original Microsoft video; Apple had Quicktime. You could have even video in these presentations as well as animation, interactivity, menus. Maybe a year after I got there, someone sent me a link and said, “There’s this program called Mosaic, you should take a look at it.” I looked at it and it was a text-based program, at that point. It wasn’t really graphical, as I remember, and you could follow some links to some chemistry or some physics or some other type of stuff that was elsewhere. I had worked in hypertext so

much and I was so busy doing Director that I said, “Yet another hypertext system. ‘Yawn.’” The first thing wasn’t much of anything. A lot of it is timing, more than anything. It must have been, ’94, ’95, ’93...

Weber: Probably ’93, it was pre the MI [Mobile Internet?] images probably.

Meyrowitz: Right, it was just a text thing in ’93. It just looked like a hypertext system and the Internet was really just becoming public. People were mostly setting up email. That was the big thing, they weren’t really thinking about links. Even though Mosaic was following links and bringing up another document; it was just text; it looked kind of like automatic FTP. If you touched on something it brought a document down. It didn’t have to say get this file name, I touched on link and it came. I don’t think anybody had the vision of what that could be back in ’93. Even Tim Berners-Lee. There was a basic vision, the Internet started going public. John Doerr who I remember was on GO’s board and Macromedia’s board just started saying, “Internet, the Internet, everything is going to be the Internet.” Around that time Marc Andreessen and his gang—this was at University of Illinois—

Weber: NCSA [National Center for Supercomputing Applications].

Meyrowitz: NCSA, which was...

Weber: They were on the campus.

Meyrowitz: On the campus of University of Illinois.

Weber: It’s an independent, yes.

Meyrowtiz: They started putting online images into Mosaic and fonts and things like that. All of a sudden, it started looking like something credible, because all the hypertext systems from the research community had all—by ’91—they were all rich with text and graphics and video and animation. The first Mosaic didn’t look like anything but then when the images got there and the fonts got there, it started looking like something. That’s when the Internet really went public. Tim’s architecture was, in retrospect, a great architecture. He distributed notes that really didn’t communicate very much except sending things back and forth. How there wasn’t any state between them. There weren’t any databases. Nobody knew about anything. It was just, ”Give me this document.” In retrospect, it was brilliant. Then they added CGI to the hypertext protocol, which was essentially a way that you could run some scripts on the server side. Again that was just a feature at that point and it became the gateway for, not just sending files back and forth, but for sending the file, running some type of program on the server and then returning results, which was the predecessor to all these applications servers that eventually came to be. That was kind of interesting. That was the first time—or one of the first times—where linking—in HyperCard and other programs, there were scripts that ran in the client to do some processing. All of a sudden now, you were following links to a distant server. The distant server, which may have had databases or other things around it, could do some computing and then send the results back. It was kind of distributed computing, in a very straightforward sense. There used to be RPC, which was Remote Procedure Call protocol. It was very low level; one programming language statement talked to another programming language statement through a

connection they had created. It was very low level and you had to do a lot of programming. Here, you just follow the link and name the script and pass some parameters and some distant computer executed and returned the results. That turned out to be the first time, when it was very easy to add scripting and processing to hypertext.

Weber: How different was a typical commercial client-server system?

Meyrowitz: The difference was that—

Weber: I don't mean the hypertext.

Meyrowitz: The processing was the same that you would do on a server. The difference was, with RPC, it was a particular procedure call that a programmer wrote. It went to a distant computer, where that computer executed this small procedure call, and then returned the results to a program. It was very, very low level. That with CGI you followed the link, it executed the whole program and the link was all done not by a program, but it was done very straightforwardly in HTML. All the links went to the same server. The same server, the CGI processor, didn't go to a particular procedure in some program. It processed that script. That script could be written in whatever language you wanted, you executed it, it returned the results. It was remote procedure calls for the rest of us, for people who were not deep level programmers. It was client-server, but it was client-server at a much higher level. In terms of multimedia and hypertext, back in Macromedia, John Doerr had been saying, "Internet, Internet, Internet," and so we figured maybe we should do some Internet stuff. We were very clever. John Doerr says something 100 or 150 times and you start doing it. We sat down with all the applications that we had; Director, Sound Edit, Authorware. Then we said how do we make these for the Internet? Right before Windows '95 came out, Microsoft really wanted something to show off the ability to have OLE [Object Linking and Embedding], which were embedded applications on the Windows system called Blackbird, which was their competitor to AOL. We took the player for Director, which played back all of the Director movies, as they were called, and we ported it so that it could actually run in the OLE inside a Blackbird. It was cool and Microsoft was appreciative. Blackbird was the predecessor to MSN when it finally came out, but it was different from the MSN of today.

Weber: It was meant to be much more than MSN.

Meyrowitz: Right, it was meant to be just like AOL with some clients, now MSN has become a web property with some backend servers, especially after they added Hotmail. At that point, Blackbird was like AOL, we did that Blackbird component. There were some demos and then when John Doerr and Bud Colligan, who is our CEO, said, "Internet, Internet, Internet," we had the idea of, "Let's take this thing that we have and make it Internet capable." At that point the Internet was so slow. Everyone had modems. The modems were 14.4 [kbps], which is ridiculously slow. A lot of the people [who] were working on Director said, "This is just a fad; we have real work to do for CD-ROMs." I said this wasn't a fad. We went out and hired a couple of people; Harry Chesley, was at Apple and had done a lot of HyperCard stuff; and Sarah Allen, who had worked for me back at Brown and had been at Collider, which was a big multimedia operating system company that came and went pretty quickly.

Weber: They were here?

Meyrowitz: They were based in San Francisco. Dave Kaiser, who was the former VP of engineering at Macromedia, was one of the founders of Collider. We hired Harry and Sarah. Harry knew a lot about the Internet, because he had written a bunch of programs. I believe they were terminal emulators and stuff in HyperCard, which was a tour de force back then working over the Internet. He did some stuff with the Internet. He might have done some interfaces with Gopher. Harry and Sarah and another guy, named John Newland, created Shockwave, which was the version of Director Playback that ran inside of web pages. We worked with Netscape. We were pushing to have an API for plug-ins. There were no plug-ins for API and John Newland worked with John Giancarlo [ph?]—that's not the right name—one of the first guys at Netscape. He was a great guy, he went on to the voice company that just got acquired by Microsoft. The voice directory company. The name escapes me.

Weber: He was one of the founding Netscape programmers?

Meyrowitz: He was one of the founding Netscape programmers. They created an API so that we could run Shockwave inside Netscape. That was mid-'95 and that was amazingly cool. Now you had hypertext in the form of a Netscape browser and then you had multimedia, running inside of that browser with animations and so forth. Harry and Sarah and John were very careful about how much data got sent back and forth. When you have a CD-ROM running in a machine with a lot of memory you don't care very much about resources. If you have a movie that has to be downloaded to a local machine from a server to actually run inside of a browser, it has to be as compact as possible. We did a lot of stuff on compaction. We did a lot of stuff on security—we had scripts using the Lingo language in Shockwave—if you had those scripts and you weren't careful, you could read the file system or erase the file system. There was a lot of work on security back then, and so Shockwave became a huge hit. In the second half of '95 was the big coming out party for Netscape, as I remember. I just remembered December of '95 being huge in terms of the Internet, Netscape taking the world by storm and Shockwave was right there in the middle of it. We were very happy. Skillful but also lucky, having the technology and being in the right place at the right time to make it happen.

Weber: Did John Doerr's relationships to Macromedia...

Meyrowitz: He was on our board. John was on our board and he turned out to be on GO's board. He's a very good guy and he was on the board of Netscape and funded Netscape, so...

Weber: He connected the dots.

Meyrowitz: Yes, John would always—this could be off the record. <*break in audio*> December of '95 was a real high point for the Internet. The first high point, of Netscape stock. I don't know if they had [their] IPO yet.

Weber: August of '95.

Meyrowitz: Yes, so they had IPO. Their stock was through the roof. Macromedia stock was through the roof.

Weber: Yes, what happened? Macromedia stock was already good.

Meyrowitz: Yes, we were seven bucks when I joined in, when we IPO-ed in October of '93. We were way up there in the middle of '95 and then it went crazy by December. It appeared that we should be doing more stuff through the Internet. The Internet was starting to get fun, and a lot of people started doing Shockwave stuff instead of CD-ROMs for educational things; there were lots of games that people created. There was the Virtual Bubble Wrap, where you could have on a Shockwave movie embedded in your web page. You'd just click on it and pop Virtual Bubble Wrap. It was completely useless, but an enormous amount of fun. There were kind of re-implementations of things, like Space Invaders or something like it and Frogger and all those other things. We had something that the Netscape browser didn't have, and so that was kind of interesting.

Weber: People could also use Shockwave back to writing to the CD-ROM or at that point it would bifurcate it?

Meyrowitz: It would bifurcate it because you couldn't do everything on the network that you could on a desktop system. You could take something that you did on the web and actually put it on a CD-ROM with a little bit of imagination. Most people didn't do that, because the CD-ROM was left for big expansive productions like ten thousand pieces of art, where you could walk through a gallery or the Rolling Stones did a Voodoo Lounge CD-ROM that had all their songs and stuff going on. That was a whole different paradigm; the kind of the small movies for the web.

Weber: The light comes on your TV.

Meyrowitz: Oh, does it? Oh, that's just a little infrared. Then the next thing that we did; our other program was Sound Edit. We said, "What can we do with the Sound Edit for the web?" Sound Editor was a way to do editing of sound, where you saw the waveforms and stereo and you could change the waveforms. You could do a fast reverb and stuff on that. It didn't make any sense for the web, but we had acquired a company called Deck-2 and there was Josh Rosen [ph?] and Motts Myerberg. Motts was an incredible developer. One day, he said, "There is this institute called Fraunhofer I read this paper and they had a new way of doing music that really compresses the music, and it uses this thing called Perceptual Coding." What it says is that humans can't really hear a full range of frequencies but a CD with a 16 bit—
<*phone rings*>

Meyrowitz: A CD digitized it at 16-bit stereo, which means that it could represent frequencies from zero to 65, 535, so 65,000 frequencies. People can't hear those, so they had very complex algorithms to figure out what people could hear more of and what people could hear less of, and they would drop out that stuff. Motts knew about that and we talked to Fraunhofer. We got a license for that and we put that into Shockwave and we called that Shockwave Audio. We had all these fantastic jukeboxes written in Shockwave with this Shockwave audio, with this MP3 streaming in. We had the first streaming music

player using MP3 on the Internet. I don't think there was anything before that. I know Fraunhofer said we were the first licensee.

Weber: There was streaming audio prior to that.

Meyrowitz: There was streaming audio with Real Networks, but it wasn't MP3. There was streaming audio and streaming video on Real Networks, but this is the first MP3. It actually had better sound quality and better compression. We called it Shockwave Audio, and we embedded [it] in Director. In retrospect, we should have just made a special player just for MP3. Maybe things would have turned out differently, but that took off for a while and there were all these record companies doing jukeboxes. You clicked on an album of the Goo Goo Dolls and the thing would stream in and that was very cool. And very functional. About a year later—and this kind of really changes the history of hypertext and multimedia and then eventually mobile—I and a bunch of people saw a thing called FutureSplash. It was an animation thing but it had hokey characters, and it didn't look that good the first time we saw it in mid '96. Later on, we saw it and it had this beautiful anti-aliasing, which meant that all of the curves were smooth so you didn't have any jagged edges and it really looked good. At that point, Shockwave had been doing very well, but it was kind of a big plug-in. It was probably a megabyte or half a megabyte. At 14.4 [kpbs] that took a minute or two to download. The FutureSplash plug-in was something like 80 or 90 KB and so it took ten seconds to download it. I started getting worried, because this had a chance of very quickly taking over Shockwave. So, I went down to San Diego with Joe Dunn— who had hired me and used to be our VP of engineering and then VP of marketing. He was VP of Corporate Development by that time, and maybe still VP of marketing. We went down and met with Jon Gay, who was the key architect of FutureSplash. We met with Robert Tatsumi [ph?], who was another engineer and we met with one of their funders, Charlie—I will have to get you his last name; I can't remember it now. He had funded Jon's previous company. In fact, Jon did one of the first Macintosh games; Future Quest. It was very cool on a 1984 Macintosh. He did that while he was in high school; while Charlie was in high school. We went down there; it was the end of the first week in December and we thought Jon was amazing. I pride myself, and Joe does, on hiring really good people and being able to tell who they are quickly. And we knew. We talked to Jon for 45 minutes, with Robert and decided we had to acquire this immediately. We told him so, told them we wanted to acquire them. It was just four engineers, or maybe even three engineers and a QA person. They were mostly engineers. They didn't want to do the marketing. They wanted to be somewhere that did the marketing. They liked us. I think Adobe was sniffing around. Microsoft was sniffing around. We said, "We really want to do this," and we closed the transaction in three weeks, from beginning to end and we had FutureSplash by the end of '96. FutureSplash—Splash had some type of trademark. We had to get a new name. Kate, my wife, and I were sitting in a corner over there. We were having coffee at breakfast time. We were just throwing names out and I think it was Kate who said, "What about Flash?" which was kind of a contraction of FutureSplash. We ran it by all those guys and ran it by the lawyers. And that became the name. So I think in March of '97, Flash came out. Flash was really the first multimedia plug-in created for the 'net. Shockwave had been created for CD-ROM and then moved over to the 'net so it had a lot of vestiges of running off of a local disk or a local CD-ROM. Flash had very small download size, and it was vector-based as opposed to bitmap-based. All of the graphics could be downloaded in a very compact fashion. Plus, they had the anti-aliasing, so everything looked beautiful. The text looked beautiful. Animations looked beautiful. People started using it and I think there was—because I remember that in 2.0 in was a basic, basic, scripting language. I think there wasn't even a scripting language. Very soon after, we hired Gary Grossman; and Jon and Gary Grossman created a scripting language for Flash and that's what made it take off. Even with a small scripting language, with a small download of a plug-in, it almost felt instantaneous, because with Active X, at least on Windows, it downloaded and installed itself virtually invisibly. On Macintosh you had to download a plug-in and install Netscape, but on Windows and

Internet Explorer, it just happened immediately. Then the content streamed down very compactly, so people were enamored with this and they could create all sorts of quizzes, and animations and cartoons. Then, they created a more complex scripting language called ActionScript, and that took Flash to a higher level. So by '98 and '99, Flash was really the way that people were doing multimedia on the web. There were a bunch of competitors that tried to knock us off, but Jon is brilliant and he figured out a way to get incredible functionality in a small plug-in. The budget that I would tell the Flash team is, "It can't grow the plug in more than 20KB for [the] entire release." Because the reason we were successful is that it was instantaneous; it was invisible. The minute that it becomes non-invisible, nobody wants that on their websites. Advertisers don't want that on their website, you know, newspapers don't want it on their website. So, keeping it small—at that point when people were still using modems—was really the key to its success besides also having great functionality. But, so then Flash just became a hit.

Weber: And the main competitor is?

Meyrowitz: There weren't any. Before the 'net, there was a competitor called M Factory for CD-ROMs. They had beautiful interface but they never went anywhere. For Flash there was not many competitors that really took off. Microsoft was pushing some type of animation or another but, you know, it never really went anywhere. I can't remember what it was but it used this weird type of program that nobody really understood. There was a product called Emblaze, but it never really took off. Flash had an open road for a while. Microsoft kept trying to do Flash killers—I don't know if they called them that—but one of these was this animation that used predicate. I think it was, predicate programming. Some computer science-y thing. Their download was mega-buggy, so that never went anywhere.

Weber: What was the revenue model for Flash?

Meyrowitz: The revenue model for Flash was sell the tools. You needed to sell the tools and every time somebody hired someone to write a Flash application for their website, there was a designer who had bought Flash, and bought the upgrades. It wasn't a model where it would scale [with] the number of users on the web, it was model that would scale [with] the number of designers on the web. At that point it was the Internet frenzy, everyone said, "You can't make money with tools. You have to make money with eyeballs, you know, just collect as many eyeballs as you can and then the money will come to you." We had been around before the Internet, and we had to be profitable. That meant that we couldn't hire numbers of people, and we always had to have better earnings than the last quarter. We couldn't afford a business model where you didn't collect any money and you just collected eyeballs and then hoped that you would IPO and get a zillion dollars and continue not to make money. In the end, it turned out that we were some of the few left standing. Now, the tools business—and for those who have great tools, is a pretty good business. Adobe acquired Macromedia in 2006, I believe. There is close to a couple of billion dollars in revenue currently in tools.

Weber: So, essentially the model didn't change much from the CD-ROM days, it was...

Meyrowitz: It was the same model, it was the same model just different medium.

Weber: And pricing was?

Meyrowitz: I think that pricing moved up as...

Weber: What does the kit cost now?

Meyrowitz: Probably in the $395, $495 range. Then there is a Studio that you buy for $1,200 that has a bunch of stuff in it. For us we have Director and SoundEdit and Extras, which is a graphics thing, FreeHand, which is our drawing program. I think it was a thousand bucks and now Adobe has a zillion bundles with Flash and Illustrator and Photoshop.

Weber: Yes.

MEyrowitz: So Flash, ended up getting MP3 in there, people used that for sound. Eventually, in about 2002, 2003, the Flash team decided to put video in Flash. It was just basic video with our basic codec; and it was kind of interesting, people started using it for video. Then in the next release, they put in a much better codec and now all of sudden you could have really good quality video. I think that it was the same codec that had Apple used. Sorenson, it was the Sorenson codec. All of a sudden Flash was what started being used for video because up till then you had Windows Video, or Quick Time, or Real Audio/Video and the problem with those it was that they were too intrusive. They put up their own interface adornments. They were large plug-ins. They had logos of their own all around. People wanted just plain video without branding from some other company and they wanted a small plug-in. They didn't want to have to download megabytes worth of the Real Player, which had a zillion pieces of stuff. The same with the Windows Media Player. So all of a sudden, people started using Flash for video because Flash ran on Macintosh's new PCs. Quicktime sort of [ran on Macintosh PCs], nobody really had Quicktime on Windows, where everybody already had Flash. We had updates, new ones would come in. Flash video started getting really huge and then YouTube started using it, and other people starting using it because it was great for YouTube. They just had to deal with the server side and they had to write a little bit of a JavaScript to embed the Flash movie in the HTML page and they had video.

Weber: And the controls are part of the Flash player basic full screen.

Meyrowitz: No, actually that's the real beauty of Flash, that there are APIs there to make things full screen, or play, or stop, or rewind, or what have you. All of these websites got to create their own controls so they could actually look the way they wanted. You know, they could have it look split. MTV could have it look cool. A comic site could make it look like a superhero video player, that was the beauty. The last version of Flash they put in a H.264 codec, which is the same codec that is used for MPEG-4 video on TV and so quality is quite impressive. The reason you don't see even better quality on the web isn't that [the] codec can't do it, it's just the bandwidth. The video wouldn't actually be too big to actually stream to any individual, it would just be too big to pay for all that bandwidth for a company like YouTube to have a hundred thousand people downloading a very high bandwidth video. That's the cost benefit; if you keep the quality a little less, it costs you less money to actually stream all that stuff to other people. That's how Flash became the real multimedia part of the web, and of hypertext. Actually there's one more thing, we kept including the scripting language at one release. ActionScript was made compatible with JavaScript. JavaScript was the scripting language that Netscape invented back in '96. It has nothing to do with Java except the name was Java, because Java had some cachet, but it was very confusing. When you talked before, what was the other competitor? The other competitor was allegedly Java. There was a Java plug-

in but, it was huge. There wasn't a scripting language and an authoring tool. You had to write, load down Java programs so it took you ten times as long to write something in Java, as it was to draw it in Flash and write a few scripts. That was the key competitor that I had forgotten about

Weber: When were they trying seriously to push it?

Meyrowitz: They were trying seriously in '96 and '97, in '98. They always tried but then by the end of the '90s, they were focusing more on using Java as a server language for application servers and for big companies. Like IBM, who is using Java a lot for backing and stuff. The other reason that Java—they put up a big logo and it took a long time to download and it took a long time to load. They had just missed what the real design targets had to be. Sometimes it's what you leave out that's more important then what you put in, which is all what Flash is all about. That's what made it successful. That's kind of the story of Flash until probably around 2003, 2004. I came back from a sabbatical and that's when mobile computing was really taking off. The question was, "What should we do?" That was pretty much the same question that was asked when we went from CD-ROM to the web, "What should we do?" This time it was, "We should get Flash on all of the handsets." The handsets weren't powerful enough five years ago to put the entire Flash plug-in. In 2008 you have an iPhone, a very fast processor that can run Apple's entire OS, or a large part of its OS with very fast graphic speed. That wasn't what it was five years ago. They were very slow processors with very small bitmap displays. We created a product called Flash Lite. We were up to Flash 6 and it was kind of what Flash 3 had been, with a little bit of scripting. It was going back to move forward. There is a theory of S-curves, where for every innovation, there is an S-curve and then the next one, innovation, starts with lower functionality, but eventually grows. You have to overlap curves, which I can't really do with my hand, you have to actually make an S and I can't do that with my fingers, because I am not a contortionist. We created a version of Flash on a variety of phones, operating systems on Symbian just as a proof of concept. Then Peter Meehan, who is our VP of sales and marketing for the Mobile Division; he spoke Japanese, he went to Japan. He convinced DoCoMo, which was the biggest cell phone company in Japan. They were the clear leaders. They had invented something called i-mode, which were kind of like web applications, but text web applications, that ran on phones and it was a big, big hit. There was a little bit of media but people downloaded all these applications the way that people are now; the way that later on, people downloaded Palm applications to Treo, or the way that people now download things to the iPhone. So i-mode was a big hit.

Weber: When did you go meet with them?

Meyrowitz: This was in 2004, something like that. I believe 2003, 2004, 2005. Peter met with and knew Natsuno-san [ph?] at DoCoMo, was one of the founders of i-mode. He had the vision, he said, "We have to have multimedia. We have to have Flash." There was another media standard that the web community had done. Oh no, it was MMS, which the telephone community had done. It was kind of crummy in terms of its graphics. He said, "You want a big hit. If we want a big hit, we have to put Flash on these phones." There were six phones from six manufacturers, and our VP of engineering for mobile, Garrity [ph?] and a whole bunch of people, they essentially lived in Japan. They got Flash Lite to run on all the phones with all the manufacturers and they were all different. We launched after a year and this became the next generation of i-mode. It became a huge hit in Japan, because the Japanese culture likes animations. You could create all these animations, create all these games so, it was a huge hit. Flash on mobile became a big thing. I had wanted to do a product called Flash Cast, which essentially pushed information to the screen, to your handset. As opposed to you having to browse, you would sign up for different channels

and while the phone was in your pocket, stuff would be sent to you. When you actually took your phone out of your pocket, there would be information for you to read. Usually, you are sitting waiting for someone in a coffee shop, waiting for a meeting to start, and if you have to turn your cell phone on, browse for information, you don't get it until the meeting starts. So you're nowhere. Flash Cast pushed all this multimedia information out there. From what I understand, real versions of it are coming out in U.S. cell phone carriers. But you know, the Flash Cast server was being actually used in Japan.

Weber: When?

Meyrowitz: I think that it was in 2006.

Weber: You know the company ACCESS [ph?], they do browsers for DoCoMo, right?

Meyrowitz: Right.

Weber: But this was straight with DoCoMo, it can run independently with Web Browser.

Meyrowitz: It ran independently of Web Browser, which is a multimedia application.

Weber: I don't know if you know Pei Wei. He did the Viola Browser.

Meyrowitz: Yes.

Weber: He works for ACCESS now.

Meyrowitz: Uh-huh.

Weber: The founder of ACCESS comes to Sunnyvale and we are going to try to interview him but that would not take care of the streaming video part of this.

Meyrowitz: Right. We were the multimedia part of it. We didn't have streaming video on Flash Lite.

Weber: Right, two different threads completely.

Meyrowitz: Right, and ACCESS did some of the i-mode browsers and then some HTML browsers. I know they did the HTML browsers for Palm at one point, and then they bought Palm's software business, which was weird, and then they sold it back to Palm, which was also weird. That's what I know about ACCESS.

Weber: This takes us back to the time you left.

Meyrowitz: Right, I left mid-2005.

Weber: Around the time Adobe acquired it?

Meyrowitz: About nine months before it got acquired. I was gone, so I don't remember the dates. It was a good acquisition, because they were very compatible and different product sets, so there wasn't very much overlap. The only overlap was the FreeHand Illustrator. There was one more big thing for the web and for hypertext. I remember we were on a plane, Bud Colligan and I went up to have dinner with Brad Silverberg, who was the guy who ran Windows '95 for Microsoft. I think that I had met him because he went to Brown, with Andy van Dam. He worked with Andy van Dam a few years before me. We went up and had a meeting to see how we could work with Microsoft as opposed to them kill us. To our credit we figured out how not to get killed. At that point Adobe had a product called PageMill that was a basic web editor, people were using BBEdit as a text only kind of programmer's editor to edit HML. It wasn't WYSIWYG.

<phone rings>

Meyrowitz: That's what editors are for. It wasn't WYSIWYG, BBEdit. You were writing a program essentially. HTML was a programming line where you look like you were writing a program. That was a big pain; you wanted the marketing person to write a paragraph or two in a web page, without having to learn backslash P, or backslash B, or whatever for bold.

Weber: Your intermediary offering was integrated.

Meyrowitz: Offering, you just opened the document and typed.

Weber: It was actually not Bernsley [ph?] but some of the later people with browsers that left everything out. Did you have any sense of that? Did you miss editing before?

Meyrowitz: Yes, we were working—we would talk to Netscape and say, "Why don't you put editing in here?" At one point they did. The problem with the editing is that—it was one of the negative or weird things about the web in that, if you are editing, you have to edit the local copy. Right? Because the real copy is on the server. You have to somehow get it, edit the local copy and then get it back down to the server. That was way beyond most mortal's ability to do.

<tape ends abruptly>

END OF TAPE 2

START OF TAPE 3

Meyrowitz: That's a problem of the Web architecture; that it's very hard to do editing because the documents are distant, and you have to make copies and put them back. Some people don't have access rights, and eventually there was a protocol called WebDAV [Web Distributed Authoring and Versioning] that got created to do this. It was hard, plus it's hard for regular users to edit the Web just because if you take a page that you haven't created from scratch, there's all sorts of stuff in there, HTML programs, Java Script, et cetera, that you don't know what to do with. It's not the easiest environment for someone who isn't a professional to create pages. Eventually, we created something called Contribute years later that was a Web editor like Dreamweaver, but it had all the stuff to push documents back and forth. It was for the novice. That worked later, and it would be nice if that were in browsers. We're kind of in a day where the browser innovation has gotten squashed because nobody wants to do anything that doesn't run across both Windows and Macintosh and maybe Linux. Microsoft isn't moving particularly fast in changing Internet Explorer, which kind of keeps the level of the browser in the late '90s, early 2000 era. It hasn't really moved that much. Firefox has created some. A lot of them reserve face things for the users. In terms of adding new protocols, it's unfortunately limited by the need to be cross-platform, and the reality that the laggard controls the protocol because you have to be cross-platform.

Weber: With intermediate, for instance, because it was a true file system—

Meyrowitz: It was a true file system.

Weber: <inaudible> permission you edited the direct <inaudible> the real file directory.

Meyrowitz: Right, and we actually even had access rights for each file that said you could have annotate only where you could just create links, or you could actually have rights to also edit the document. So that meant you could—

Weber: Right, because you were storing a link in a central server you could put the anchor in without altering the target document <inaudible>.

Meyrowitz: Right. We went and had dinner with Brad Silverberg and asked Microsoft not to kill us for that quarter. Then on the way back, we had this product called xRes that was a Photoshop-like program. It had a lot fewer features than Photoshop, but it had this one cool feature where it stored a file in increasing depth or resolution, so you could bring up on a screen the full picture at small resolution very quickly. As it drilled down it would give you more resolution, but it turned out that that was a feature. We did an analysis, and we were so far behind Photoshop in terms of the features that I thought this was going to be a lot of work to always be behind, because we're smart, but some of the people at Adobe were smarter, and they have probably more people. We're sitting on the plane going home from Microsoft, and Bud, our CEO, said, "So, you're happy with all this stuff. Would you like to be doing anything different?" I said, "I don't want to be doing xRes, and I really think that we should be doing a professional editor for the Web that all of the professionals in the Web space would use." The analogy I used was there was Microsoft Word, but when people started doing technical publications, they used a product called FrameMaker. FrameMaker had all the nuance that Word didn't in terms of half points and doing lots of different columns and doing

footnotes and tables and being set for printing manuals. It turned out that, interestingly enough, Joe Dunn, who had hired me, was our VP of engineering when he hired me and then moved on. He had been a director at Frame. Art, our Director of Customer Service, had been at Frame. One of our sales guys had been at Frame. I was talking to Travis Huck, who was our sales guy, and said, "Who are the best people from Frame?" because if you're going to do the Web equivalent of FrameMaker, with all the nuance, get somebody who--

Weber: And it's all SGML basically.

Meyrowitz: Yes, and knows how to do this. I don't think Frame was SGML to begin with. It might have been later.

Weber: I mean conceptual <inaudible> segments.

Meyrowitz: Yes, conceptual, so let's get someone who knows all that nuance. Travis gave me some names, and he said, "One of the guys, he's young, but he's great. He has a lot of energy, and he's Kevin Lynch. But he left to go to General Magic," which is a company started by Bill Atkinson and a bunch of others. I learned about Kevin a little bit. He had done the Mac port of FrameMaker. I think it started on Sun workstations, maybe Windows. I can't remember. But he did the Macintosh port, and—

Weber: Yes, it was Sun.

Meyrowitz: It was Sun, right?

Weber: Well, it was Unix and then Sun.

Meyrowitz: It was Unix and then it moved to—

Weber: Yes, it got out of the Mac in '88 or '89.

Meyrowitz: Well, Kevin was the guy who led that port and did a lot of the port. I kept calling Kevin and at that point, General Magic was in its downward spiral. When it went up, they were [a] doing programming language, and then Steve Perlman was off doing something else, and they were doing hardware. Bill Atkinson was doing one thing. Steve Perlman was doing another, so they were—

Weber: But wasn't that for wireless devices?

Meyrowitz: They had wireless devices, again with a modem. Anyway, I called Kevin, and I kept having lunch or dinner with him. He kept saying, "I really like it at General Magic." I liked him. I thought he was really bright, and he had a great design sense. He had a really good understanding of the nuance of

FrameMaker. We made him an offer that he couldn't refuse. <*laughing*> He joined us first as Director of Advanced Technology or something like that. I said to Bud on the airplane, "I think we need to do this professional editor, and I think I want to get Kevin to do that." That was kind of the genesis of Dreamweaver. I talked to Kevin, and it really wasn't a very long meeting. It was probably half an hour or 45 minutes. I think the description was, "Do you want to do this product that's a professional editor that just goes above and beyond PageMill by a couple of orders of magnitude that has all nuance of FrameMaker?" He said, "Yes." I said, "Hire people who are really good, and let's get it out in a year." He did; he hired some great people. He had a great design sense. They knew about—

Weber: When did it officially start?

Meyrowitz: Geez, I think '97. I think in the beginning of '97. They had a really nice user interface, but the innovation that they came up with that really made Dreamweaver catch on was what was called Ground Trip HTML. All of the other WYSIWYG editors, they'd bring in an HTML text, which all these programmers, who were some of the people doing the original HTML pages and Java Script stuff, they'd format it with the right spaces and parentheses and just how they liked it. They'd bring it into a WYSIWYG editor who would make some changes then would write it out in its own format and rack up the formatting for when the programmer wanted to go in with the programming editor as opposed to with the WYSIWYG editor. Sometimes they wanted the marketing person to just write the stuff, and the WYSIWYG editor save it and then give it back to the programmer. Then the programmer would get all upset because all the formatting was gone.

Weber: <*inaudible*>

Meyrowitz: The Dreamweaver team went and talked to people about why the professionals don't use WYSIWYG editors. Pretty much to a person they said, "Because it wrecks our format, or it adds extra HTML. It takes out stuff, and it changes the HTML commands, and it adds a lot of extra stuff that we don't need and makes the files bigger. They said, "It sounds like a feature—the nuance that professionals need is round-trip HTML where whatever you read in from a text file, you don't mess that formatting up at all." Whenever people edit, you just put in—

Weber: What they <*inaudible*>*.*

Meyrowitz: When they edit it, and when you write it out, it doesn't have any more stuff in it than what was there before the editing, and it has all the same formatting. That's the thing that made Dreamweaver take off.

Weber: Okay, that's interesting. Yes, I didn't know that.

Meyrowitz: So that's kind of the key.

Weber: None of the other—and there was what? Netscape Gold was not—didn't <*inaudible*>*.*

Meyrowitz: That didn't do that. PageMill made a big mess; not a big mess but a mess. It didn't support a lot of the HTML commands in a WYSIWYG fashion. Dreamweaver supported, as part of the nuance, it supported most of the HTML commands in a WYSIWYG fashion. There was another one that Adobe eventually acquired that was in Germany.

Weber: Oh, <*inaudible*>, Germany or there was a French one, wasn't there? Griff? That's <*inaudible*>.

Meyrowitz: I'm pretty sure it was Germany. They were pretty good, and they came up after us. But they didn't have the round-trip HTML as good as they could, and Microsoft had something. Well, they had the one that they eventually shipped with Office, which was called—

Weber: FrontPage.

Meyrowitz: FrontPage. They acquired that from a company in Boston. That was the leader. That was coming up in the middle of Dreamweaver. That was the one that we were most—

Weber: Competing.

Meyrowitz: —competing with and afraid of, plus this other one from Germany. But then Microsoft acquired FrontPage. The founder left, and they couldn't figure out whether they wanted a professional editor, which is what FrontPage was, or a user editor. They created a hybrid that was neither here nor there. FrontPage was eventually not a professional editor, so it never really competed with us that much, because they made it much more into a user editor. It took them years to put it in round-trip HTML, and by that point, no one really cared. Plus a lot of the Web developers used Macintoshes, so FrontPage didn't run on Macintoshes.

Weber: Oh, not at all?

Meyrowitz: I don't think it did for awhile. I know Dreamweaver did from the beginning.

Weber: Then the quote/unquote Web editor in Word, the pro one, I mean that was—Microsoft intended to be the kind of user <*inaudible*>.

Meyrowitz: Yes, that was—I don't know that many people who even used it. They had a way to take Word and write out HTML and put it on the Web, but it wasn't really—well, at least by our market, which was the—

Weber: Professionals.

Meyrowitz: —professionals.

Weber: So you pretty quickly got most of the professional market?

Meyrowitz: Yes.

Weber: I remember it just sweeping in.

Meyrowitz: It was the end of ’97, as I remember. ’95 was the first time the Web took off. Then ’97 and ’98 were when it really was at its peak, ’97, ’98, ’99. There were a zillion companies funded by venture capitalists, and they would buy everything by the boat load. Dreamweaver was bought by the boat load. Aeron chairs were bought by the boat load, spare no expense. The more money you were losing, the better you are. There was enormous amount of sales of Dreamweaver in those years, so we were happy. When the bust in 2000 hit, everyone was affected. We weren’t affected as much, because we still were selling a lot of Dreamweaver, and we had all these other products. So we survived way better than the companies that ran out of money and didn’t have revenue. <*laughing*>

Weber: By the, say, early 2000s, what percentage was Web versus still traditional Director-type stuff?

Meyrowitz: By 2000 the Web probably was at least half of our business. We had other products; Authorware, which was another hypertext system for courseware, for doing educational software, a lot of business education rules and regulations and HIPAA [Health Insurance Portability and Accountability Act] and fire prevention and harassment.

Weber: CBT [Computer Based Training].

Meyrowitz: CBT kind of stuff. We had FreeHand, which was a competitor to Illustrator, and it was actually [a] pretty decent revenue stream at that point. It was very popular, and Illustrator hadn’t really pulled away at that point. We had SoundEdit. We had enough—it was probably 50/50. Director is still sold, and Shockwave is still used. Dreamweaver is probably, in order of magnitude, more now and Flash is probably, in order of magnitude, more.

Weber: You really had become a Web company essentially.

Meyrowitz: Right.

Weber: Can I ask why you left?

Meyrowitz: Just I had done this for 11 years, and pushed out an enormous amount of products. I was managing all these things, and I enjoyed it. I thought I was pretty good at it. I hired a bunch of people and put out a bunch of products that were hits. I’d like to think that it wasn’t an accident and that I had something to do with it. A lot of people had something to do with it. It just wasn’t me, but I’d like to think that I contributed some overarching rhyme and reason to some of the stuff. From the time I got there to

the time I left, I think there was Director 5, Director 6, Director 7, Director 8; Dreamweaver 1, 2, 3, 4; Flash, 2, 3, 4, 5, 6; some releases of ColdFusion. We started Flex, which is now becoming pretty popular for rich applications; a zillion versions FreeHand; Flash Lite; Flash Cast, and at some point you get tired of waking up at 3:00 in the morning thinking about which product has problems. When Apple does its PR , it makes it sound like all products get done in a pristine, almost godlike fashion. There are never any problems, and everything is beautiful. But the reality of software is that it's kind of Whack-a-Mole, that there's always a problem with one team, and it's always a different team. It could be a quality problem or a user interface problem, or they're behind schedule, and they won't save it, or somebody on the team isn't staying. There's always something that you have to spend all your time on; the thing that's failing or that isn't working, as opposed to—if all my days were spent with the teams that were really doing great, and we were just innovating and innovating and creating great stuff, it would be fantastic. I was President of Products at the end for a zillion years, so I was overseeing all these business units. My job was just dealing with—

Weber: The fires.

Meyrowitz: —the fires, and so at some point I decided, "I know how to do this. I feel comfortable that it wasn't an accident. There's a lot of people that I hired that are really good, and they would like to have a crack at this." It was a good time to leave, because I had been doing this stuff from freshman week, when I had the seminar with Andy. That's 1977 to 2005, so that's 18 years. All of those years were either working on projects with Andy years, which were 18-hour days, or they were Intermedia years, which were also 18-hour days, six days a week. If you double the 17 years, that's 34 years. When I left Macromedia, I was 43-years-old, so 43 and 34, 78. It's time to retire.

Weber: What was it like working with Andy?

Meyrowitz: Oh, working with Andy was great, and we're great friends. I was his right-hand person in the department for the first three years when I graduated. I wrote proposals; I helped raise funding, and I did Window managers. I helped build our new building. I knew everything that was going on, and so I did everything for Andy. Andy is great, because he's visionary. He causes everybody to work really hard, and he'll push you to your limit to do good work. Sometimes he can be infuriating, but it's infuriating for the right reasons, because he'll tell you what you're doing wrong. You may not want to hear it, but it's right, and you learn to take the criticism and deal with it. Plus, people worked like crazy doing all these projects. I was an undergraduate doing this Window manager and getting it done and showing it to the guy who heads the Office of Naval Research and so it was kind of heady, and it helped you learn how to do this in industry.

Weber: You had a very unique background being grounded in Andy's work and also hypertext, which very few people had any experience with. Do you think that those early experiences really shaped what you did later?

Meyrowitz: Yes. I occasionally think about how lucky or charmed I was, because I was really into journalism in high school, both writing and layout. Then I went to Brown and asked about automated newspaper layout and started working on hypertext. Then the whole notion of hypertext at Brown was the linking but also to have portable machines that could be on aircraft carriers for repair manuals. Then I

went to GO, which was doing mobile stuff. Then I went to Macromedia, which was doing multimedia, which was one of the systems that we did at Brown, the electronic document system was very much like the multimedia stuff that Director did. Then we started working on the Web, and the Web is just this aggregation of writing and page layout and multimedia. Then with handsets and laptops and wireless, it's mobile computing. It all built on itself, and to a point where sometimes things would just look obvious, these big innovations that people would write about in the press. Like Pen Computing, big thing in press how it's going to change. Nobody ever said, in all the stuff that happened last year or whenever Microsoft announced the tabloid computing initiative, nobody even mentioned GO or Grid or whomever. They used all the same marketing language; I could swear they used some copy that I actually wrote 15 years ago.

Weber: Did you write marketing copy?

Meyrowitz: No, mostly I did engineering, but with Mike Homer, I wrote marketing copy because I knew how his stuff worked. I was a writer, and I—

Weber: But "The pen is the point," Microsoft didn't quite use that.

Meyrowitz: They didn't use that just maybe because they forgot. There's an interesting story about that, I'll tell you off the record. So, what was I talking about?

Weber: <*inaudible*> Oh, just how much the—

Meyrowitz: I'm kind of a computer history buff, at least of hypertext, and I think there's an archive of mine at Brown, because I gave them all of my hypertext documents. When I left, I gave them—

Weber: When did you leave?

Meyrowitz: I left in '91.

Weber: I went there in '96 to interview Andy. He had just a little museum of machines.

Meyrowitz: He had a museum of machines, but there're at least two cartons of documents that I gave Brown of the history of hypertext with all these hypertext machines—

Weber: And do you know who <*inaudible*>? Is it in the CS Department <*inaudible*>?

Meyrowitz: No, I think it's in the Brown University library somewhere.

Weber: Okay, because that, I mean that's almost an off-the-record, just because I think it's of interest generally.

Meyrowitz: Right.

Weber: I do want to ask you about some Brown <*inaudible*>. You're one of the few people that had exposure straight from the '60s pioneers in hypertext and graphics.

Meyrowitz: Yes.

Weber: Andy was a big graphics pioneer as well.

Meyrowitz: Right .

Weber: Straight through to now in mobile computing. You think that the things you learned prior to '87 or so did <*inaudible*>?

Meyrowitz: Oh yes, all that stuff did, and it turns out the vision has always been the same. It's just that every five years or so technology changes so that you can do something different than you did before. When we did the Window manager, it was Ramtek display with 64 KB of memory, and sent display lists down to draw things. Having huge images wasn't the biggest thing in the world. Then, workstations came along and you weren't sending things down—it was a Unibus for the VAX-11/780, a Unibus connected the Ramtek to the VAX, so it was a small pipeline. Then five years later, workstations came out, and you had a direct pipeline to a master screen. They were black and white to begin with, and so that was a problem. Then screens became color, so you could do different things. There were local area networks for the Apollos, so you could start doing things over the networks. Then you could start distributing file systems, so you could do some of that. Then there was these tablet computers and some laptop computers. There was the Kaypro and the Osborne in the late '80s, and those were great except the screens weren't that good and the battery life wasn't that good, and they were heavy. But you could do some things. All of the sudden there are color displays with an operating system like Windows or Macintosh that has Window managers. Then the Internet comes, and e-mail. In their business, it's fast, but at home it's slow. You're constantly doing the same vision but changing it based upon that technology. The hypertext vision is in Ted Nelson's book, and it's pretty much—

Weber: <*inaudible*>

Meyrowitz: The Web, though he was much more about non-sequential writing as opposed to the Web, which is sort of a publishing system.

Weber: And then Microsoft as well.

Meyrowitz: Right. The vision has always been there, except you keep crafting the hypertext or the multimedia to be different. You do it on portables, and then mobile phones come along, or CD ROMS come along and do that. The Internet comes along, and then mobile phones come along, and you just change the direction a little bit.

Weber: Where would you like to see it go?

Meyrowitz: That's a good question. I would like it if the stuff wasn't so heavyweight, that you didn't need to have all of these very powerful and still somewhat expensive computers with full-fledged operating systems and then all this HTML stack. I'd love it to be the way that people buy telephones now; that you could buy Web tablets with keyboards for $99, and you could use them everywhere. I also —would like to see what happens between—the other thing that I want is a nationwide network with the speed of Wi-Fi. That's the clear thing that you want, because Edge and even the current 3G AT&T stuff and the Sprint Network, which is even a little faster, it's still too slow.

Weber: How about Asia?

Meyrowitz: Some parts of Asia have faster in-house network speeds. They'll have 100 megabits to the house, largely because they didn't have the [legacy] infrastructure, the cable infrastructure that people are using in the houses are from the '60s and '70s when US cable companies laid coaxial or DSL, which were on copper lines that were done—Alexander Graham Bell laid them, and they [the Japanese] just laid fiber. I think the speeds are kind of the same in Japan and Asia in terms of the wireless stuff, maybe a little bit faster. But they're using the same—

Weber: It's still slower than you would like to see.

Meyrowitz: Still slower than I would like. I want to have Wi-Fi speeds everywhere, so if WiMAX takes off from Sprint and Clearwire. Is that their name? They're actually joining forces. That will be great. AT&T says their HSPA Network is going to get up to 10 or 20 megabits. I haven't heard what Verizon is doing, because Sprint is doing their WiMAX, and that's one protocol that's the CDMA. AT&T is on their different networks, the CDMA network.

Weber: AT&T is GSM.

Meyrowitz: GSM, excuse me.

Weber: *<inaudible>*

Meyrowitz: Sprint is CDMA. AT&T is GSM, so they can't use the same wireless protocols. Verizon is also CDMA, just like Sprint, but I don't know what they're doing in terms of high-speed networks, but we'll see. Some people say WiMAX is going to work. Other people say it isn't. WiMAX is supposed to have just about Wi-Fi speeds and very high bandwidth, very high speed. You won't even need Wi-Fi in your house.

You'll just use WiMAX. I'd love that, because I have a Centro, a Palm Centro, which I've had for a long time, and I can type very fast on it.

Weber: You don't miss the pen [computing] days?

Meyrowitz: I don't miss the pen days that much, because I can type very fast. I can use a five-key keypad fast. Whenever you want a point, there's a pen on the Centro, or you can use your fingers. I carry one of those around, because the Sprint network is pretty good, and I can get phone and data all the time. I had an iPod Touch, and it works great in the house with Wi-Fi. When I take it out, there are no Wi-Fi hotspots I can use, so I don't have connection. Plus, I don't believe that the typing on a virtual keyboard, the way that Apple has it, is any better than handwriting recognition.

Weber: Oh, it's considerably worse.

Meyrowitz: I find it infuriating. I was going to buy an iPhone and I tried everything out on my iPod Touch. Just had an acid flashback to the days of doing handwriting tests—

Weber: At GO.

Meyrowitz: At GO, compared to a keyboard. It was something like eight times slower with handwriting than a keyboard and the same with the iPhone or the iPod Touch. The typing is so slow. Amazingly enough, I'm in the middle of San Francisco, and I can't get an AT&T signal.

Weber: This is the dead zone, you know.

Meyrowitz: Yes, because of Sutro Tower.

Weber: I don't know. Every time I drive up Market Street, I go through a dead zone.

Meyrowitz: Yes, that drives me crazy. I think an iPhone with a keyboard would be way better—if Apple wanted to make money in business, I'd put a keyboard on it, because I don't see that many people who have BlackBerrys, or Centros or Treos bannering the keyboard, the thumb keyboard. I really can type fast on the thumb keyboard. I think I have carpal thumb syndrome.

Weber: So you see mobile as the future and it sounds like interface is one of the major issues?

Meyrowitz: Yes. One is I think thin. These tabloids are—which this is actually thinner than most tabloids—way too big, way too heavy. I think the Kindle is pretty nice, but it hammers on electronic book. The E-Ink screen is nice, because even when you turn it off there's an image. It's lower power than today's LCD screens. But it's not color; it doesn't have very much resolution. This is something else that I would

like; batteries that last longer. The other thing, screen resolution isn't used properly. It's used to fit more things on the screen. It really should be used to make everything on the screen look better. We need much denser screens. Instead of 150 dots per inch or 120 or 70—which is what's on laptops—if you had 400 or 500 dots per inch, stuff would look like a magazine page. Now, it still looks like a computer. I want it to look like a magazine. The reason the iPhone looks so good is because they have very high dots per inch.

Weber: Oh yes, higher than normal.

Meyrowitz: Higher than normal, higher than you'd see on a computer screen. I mean, just think. It's 640 x 480 and just think about putting four iPods together. It's much denser than a laptop, and four of those would be 1280 x 1024, much smaller area. That's where I think we're going wrong. I don't think the display technology has really changed that much. I think it's getting bigger, but they haven't really gotten much denser, which gives you much better anti-aliasing, much better looking characters, much better looking everything.

Weber: Then you see laptops as staying or would like to see everything go?

Meyrowitz: Oh, I think laptops will stay. I think it's more likely that you'll carry around a keyboard, and you'll pull out a thin film display, as opposed to people carrying around a tablet without a keyboard, because it still is pretty fast to type. It's not an accident that the typewriter kind of became an institution from the early typewriters of the 1800s, where they were designed so the keys didn't jam. There are various innovations. Doug Engelbart, in his NLS system, besides inventing the mouse, he had a chord keyboard, which was like five piano keys. If you knew the right combinations, you could get all of what we now call the ASCII character set. It might be nice to have that, but I don't see that happening. Every once in awhile I read that somebody has created something similar to that. I think it's harder for people to learn double key or triple key combinations than it is to do one character at a time, which is what typing is, even though it's fast. Otherwise, you have to do literally chords.

Weber: Any other sort of lost futures over the breadth of computing history that you've been a part of, things that you'd like to see come back or the lessons from the past?

Meyrowitz: There are much fewer innovations than there used to be, just because it used to be a green field, a lot of space for everybody. Now it's an incredibly crowded marketplace, but I don't see lots of innovation. One innovation I really liked, and it was years ago now, was PointCast. Do you remember PointCast?

Weber: Yes, I remember the name.

Meyrowitz: PointCast was a screen saver. You slapped in some preferences, and it downloaded all sorts of information, weather, news, sports. You would just go about your work, and when you went away and came back to your desk, the screen saver would have all of these pages that you could browse through. They were very pretty, and so it was real push technology as opposed to me having to browse. I thought it

was great, and it died not because it wasn't a great interface. I thought it was great. It died because (A), the client code, the software code—the software that ran in the screen saver kept crashing people's systems just the way it was architected; everybody who signed up for PointCast, and it went like wildfire across organizations, everybody got a separate feed. It just totally took down everybody's network. I miss that kind of push technology. You get it a little bit now with RSS feeds. But it would be nice to have more multimedia RSS feeds where the information just floated by. My catch phrase is, "I don't want to have to find the information. I want the information to find me."

Weber: That's good. Anything else you'd like to [add]?

Meyrowitz: It's harder to see what's missing. I think what's missing more is the ubiquity. One thing I'd like to see is—I don't understand how any humans can set up their own—who don't have computer science degrees like me—can set up networks in their house. It is not like plugging in the telephone. It is having to understand gateways and subnet masks and DNS and firewalls and it's in sync. Then if you want to open up some of your systems to the outside world, so you can have a Web surfer or dial into them, you have to fool around with firewalls and create virtual static IP addresses. One of the things I do is just for fun I set up friends' networks, just because it keeps my mind going. I find this stuff to be very arcane. All the companies, they throw in a lot of junk on the routers and access points and so forth, but they don't ease use. If it's going to be like the telephone, the network is in the stage right now where if you compare to phones, it's like when you called Centro, they plugged in something into a real switchboard. It's kind of—

Weber: Now you have to plug into <*inaudible*>.

Meyrowitz: Now, I have to plug things into the switchboard.

Weber: Are you looking at doing other ventures?

Meyrowitz: I have looked at doing one venture that I can't mention. I have to see if and when I have the energy to do it, because it would really change peoples' lives, and it's something that everybody would want immediately. It will require a lot of cooperation of businesses and banks and financial institutions. If you want to do that, you have to have lots of energy and probably be younger than I am. I'm 48, which is like 73 or something.

Weber: You've paid your dues.

Meyrowitz: It's just not that I paid my dues, it's just that I've had all these negotiations with big companies. For years the advertising agencies said, "We're not going to use Flash. We're not going to use Flash. We're going to use Flash. Not everybody has the plug-in." We'd show them charts that say 97 percent of the people have the plug-in. Even when we do a new one, in six to nine months, everybody has the new one. It was like more than a billion downloads of Flash, which is kind of cool. The advertising agencies would just say, "We're not going to use it. We're not going to use it. We're not going to use it." You'd have to just go and just constantly push and push and push. Then eventually you win, and they start using it for advertising all over the place. Then they come back to you and say, "Why didn't you tell us to

use this?" What I'm thinking of doing would be the same thing, that everybody is going to have an excuse not to do it. You have to go through these 40 years in the desert, and then you'd get back to The Promised Land. Everyone says, "Where were you?"

Weber: Okay.

Meyrowitz: Good.

Weber: Did you know the fellow that died recently?

Meyrowitz: Oh, Randy [Pausch]? Yes.

Weber: <*inaudible*>

Meyrowitz: I was his TA. He worked on the electronic document system when I was working on the Window manager. He did the graphics anchor for that. He was a great guy, and it's very sad that he's gone. Though I think his talk is fantastic. Brown called me up when they were writing an article about him. Besides saying that he was very energetic and a great teacher, I thought that every university should have people watch that video on freshman week, because that kind of stuff is never taught. It used to be taught in homes and churches and religious institutions, and now no one teaches how to live your life anymore. To me this was once you see that, it changes the way you go to college. It changes the way you work. He left a legacy, and he went out the way that he wanted to. It's a shame, but his kids and wife are provided for, so they did the best they could do.

Weber: Well, thank you very much.

Meyrowitz: You're welcome, and if I know any more history, I'll tell you.

END OF INTERVIEW

