



Desktop Publishing Pioneer Meeting: Day 1 Session 3 - Technology in the 1970s

Moderators by:
Burt Grad
David C. Brock

Editor:
Cheryl Baltes

Recorded May 22, 2017
Mountain View, CA

CHM Reference number: X8209.2017

© 2017 Computer History Museum

Table of Contents

TECHNOLOGICAL FOUNDATIONS IN THE 1970S.....	5
USER ACCESSIBILITY AND ACCESS.....	7
SOFTWARE DEVELOPMENT.....	8
ENGELBART DEMO AND SCALING AND COST FACTORS.....	9
USER INTERFACE DEVELOPMENT.....	11
TOOL DEVELOPMENT AND EXPERIMENTATION AT XEROX PARC.....	14
ALTO, DOVER, AND OFFICE PRODUCTS.....	15
OFFICE PRINTING AND PUBLISHING.....	17
GINN AND XEROX PARC.....	19
FONT DESIGN.....	19
OPTICAL PRINTING TECHNOLOGY.....	22
WYSIWYG CAPABILITIES.....	23
CANON.....	23
BRAVO AND GYPSY.....	24
PRESS/INTERPRESS AND THE LAUNCH OF ADOBE.....	26
OFFICE VS. NEWSPAPER SYSTEMS AND USERS.....	27
NEWSPAPER ENVIRONMENT IN THE 1970S.....	28
NEWSPAPER INDUSTRY AUTOMATION AND ADVANCES.....	29
BIRTH OF ATEX.....	30
ATEX DEVELOPMENT.....	32
TECHNOLOGICAL FOUNDATIONS OF ATEX.....	34

Desktop Publishing Meeting: Session 3: Technology in the 1970s

Conducted by Software Industry Special Interest Group

Abstract: Session 3 of the Desktop Publishing Pioneer Meeting focuses on desktop publishing technology in the 1970s. The session begins with a survey of the technological foundations that made the widespread introduction of desktop publishing products in the 1980s possible. The participants speak about the Engelbart demo and its influence on later developments in the field, particularly in terms of office product solutions at Xerox PARC. An extended discussion focuses on the differences between the work and developments that occurred on the West and East Coasts and in the office and newspaper industries, respectively. The session concludes with a thorough discussion of the need for and resistance to automation in the newspaper industry and the factors that led to the rapid development of the Atex system in the 1970s.

Participants:

<u>Name</u>	<u>Affiliation</u>
David Brock	Moderator, CHM Center for Software History
Burton Grad	Moderator, SI SIG
Chuck Bigelow	Typography
Paul Brainerd	Aldus
Liz Crews (nee Bond)	Xerox PARC and Adobe
Chuck Geschke	Xerox PARC and Adobe
Steve Kirsch	FrameMaker
Donald Knuth	TeX
Butler Lampson	Xerox PARC
Lee Lorenzen	Ventura
Martin Ruckert	TeX
John Scull	Apple laser printer
Jonathan Seybold	Rocappi, Seybold Newsletter and Conferences

John Shoch	Xerox PARC
Bob Sproull	Xerox PARC
Larry Tesler	Xerox PARC and Apple
John Warnock	Xerox PARC and Adobe
Richard Ying	Atex
Thomas Haigh	Historian, University of Wisconsin/Milwaukee
Matthew Kirschenbaum	Historian, University of Maryland
Dave Walden	Historian
Ann Hardy	SI SIG
Mike Humphries	SI SIG
Doug Jerger	SI SIG
Ed LaHay	SI SIG
Hansen Hsu	CHM, Historian
John Markoff	CHM
Paul McJones	CHM, Software Preservation Group
Len Shustek	CHM, Chairman
Dag Spicer	CHM, Historian
Marc Weber	CHM, Internet Curator

Technological Foundations in the 1970s

Burt Grad: Let's start by going back. Tell me what each of you consider the key elements that were necessary in order for desktop publishing to be economically and technologically viable in the 1980s. What did there have to be in place to make for a successful business proposition? Let me just go around the table. What are your thoughts on the key elements?

Chuck Bigelow: One was the concept of the mathematical description of character outlines because, as Liz Crews previously pointed out, Xerox was originally making bitmaps or rasterized elements. It is a font issue.

Paul Brainerd: The microcomputer laid the foundation, taking the minicomputer and miniaturizing it and the whole concept around Intel and creating the mass storage devices that went with it. We needed that 100x breakthrough in hardware with the computer itself.

Grad: What you're saying is that having a machine of that size and cost was critical to making this economically feasible.

Brainerd: Yes, correct.

Liz Crews: I think that it was critical to have a page description language that allowed people to have an opportunity to have device-independent output and to also be able to think beyond the size of an 8-1/2 × 11-inch piece of paper. If you didn't have a page description language that allowed you to do that, you really weren't going to be able to publish serious material. This may be usurping what John and Chuck would have to say.

Donald Knuth: Well, I guess just the fact that a digital device would be able to make a decent looking output was key.

Martin Ruckert: I think user interfaces were important, not just systems for computer scientists.

Matthew Kirshenbaum: I was doing desktop publishing on my Apple II by 1985 or 1986. I was 14 years old.

Grad: The question is what were the fundamental building blocks that we needed to have in place in order to do desktop publishing in the 1980s? What did we need to do in the 1970s to make that feasible?

Steve Kirsch: Personalized access. Easy access. Affordable hardware. Essentially a PC that could give you WYSIWYG and that you could print it out. You had disk drives that could store enough information and networking. All those were in place.

Lee Lorenzen: I'd say the graphical operating system was key so that you had a core set of functionality that the app didn't have to recreate from scratch each time: access to fonts, high-quality printing devices or screens, mouse input, menus, or dialogues—all that stuff. It was hard to do an app when you had to do all of that, which was really operating system layer stuff, plus your application. It left things easier for the app writers.

Chuck Geschke: Laser printers that a single person could lift and move.

Grad: That seems to be a very critical element. How about you, John?

John Warnock: Communication infrastructure. The growth of the Arpanet during the 1970s was really important. The Alto style machine was ambitious for its time, but as the later machines at PARC came along like the Dorado and the Dolphin, you got color and that forced the device-independent issues to come to the surface.

Grad: So, you think black and white versus color was a significant element? Or was that something separate?

Warnock: Not at the super beginning, but it obviously was a place where the industry had to go. The software, user interfaces, the mouse, the keyboard, the ability to interact with the screen—you had to have all of those components. When we started Adobe, Sun Microsystems really was the only game in town that was starting to build terminals, but they were very expensive.

Butler Lampson: As John Scull said this morning, you needed all the things that we built at PARC in the 1970s, and you needed them to be commercially available at a reasonable cost. The only thing I would add to that is what John just said about color. We made a very explicit decision not to do that at PARC because we thought it would be too expensive.

Bob Sproull: To me, for a variety of technical and economic reasons, it's the ubiquity of raster imaging for input and output, which happened to have the lovely consequence that it was oblivious to the content that you were imaging. It could be image, it could be pictures, it could be drawings, it could be text, it could be whatever. It was up to the resolution of the renderer and, of course, your eye. When I said economic, what I mean is that there's a reason that raster is cheaper than calligraphic and eventually that dominated the products and so forth followed. That was a very central piece of the infrastructure of all of this.

User Accessibility and Access

Dave Walden: All of you know more about the commercial desktop publishing world, but I think we were doing desktop publishing in a primitive sense way back as soon as we got to sit interactively at computers or computer terminals. I think there was a cultural change that was a long time coming and needed to be supported by all the hardware that the various people have mentioned. The whole idea was that an individual should get to be able to use a computer.

Thomas Haigh: I think people have identified all the obvious technological prerequisites, and I'd be curious to think if we could pin it down more specifically, maybe with say a megabyte of memory and an 8 megahertz, 16-bit processor. But pulling back slightly, let's not forget that we need to have a community of users out there who want to do their own page layout, who are interested in doing it and who can afford to buy the technology. Remember, all the kinds of infrastructure and cultural work that would go into constructing a new application category—the work of magazines and evangelists in propagating it, the labor of producing the phrase "desktop publishing." Did any of you come up with it? Where does that come from? Spreading the concept and building the demand for it is just as important as having those technological building blocks.

Grad: A marketing view.

John Scull: I think we've already talked about the technology underpinnings. I think what came out of Xerox and the whole graphical user interface started to permeate, and then the Apple Mac obviously took it the next step, and to some extent, the Lisa before it. That coupled with the democratization of it in that there is actually a sales channel to be able to bring it to the masses, but the price and ease of use had to be there. Again, everything has to come together. Once it all came together so it could get out to people, at a price that they could afford in a transformative way, then you had a chance to really have something explode.

Larry Tesler: I think we've named a lot of technologies and we've talked about when they all came together. There's two ways to interpret that. One is, by just coincidence, they all got finished the same day and we started a new industry. But there were several parts already pretty far along, and other parts that weren't very far along at all. There were people who had some need or vision that could see, "If we just can get that done, if you can get a reliable, affordable, lightweight laser printer, for example, then we could have the whole thing." A lot of people were aware that we were getting really close. This could be where Seybold Publications and Conferences come in, and this is where Engelbart's demo, which opened everybody's eyes, came in. That is another way to look at it.

Paul McJones: I don't have much to add except a data point of the awareness thing. In 1976, the *Popular Electronics* issue came out with the Altair 8800 on it that inspired Bill Gates and Paul Allen. I heard about that. I had to drive all around San Jose to find a copy of *Popular*

Electronics in 1976, even though these guys were over in their lab with their Altos. Of course, a decade later, people would open up their wallet and buy the stuff. You can sort of see this awareness and the fear factor of computing dissipating over that period.

Grad: So, culture was also a factor.

Software Development

Jonathan Seybold: This has been a very good list of the prerequisite stuff. Let me just sort of talk about what happens, I think. You had what had come out of Xerox PARC and that embodied most of the platform we're talking about here. But there were two key things on top of that that I think pushed things over the top and made the difference. The first was PostScript. PostScript was absolutely revolutionary because it was device-independent and because it dealt with text and graphics—the entire page, if you will. We didn't get to full, high-quality color images yet, but that was going to come. Early in this process—I don't recall the exact date, but it was in the 1980s—Steve Jobs called me. This story's crucial because what was missing at that point was software that could take advantage of what PostScript and the Mac could do. There was none in development except for PageMaker. That came out of the newspapers. That came out of a tradition where you needed to do text and graphics, you needed to do page layout. When Steve Jobs and John and Chuck first showed me the LaserWriter, I told them, "You've turned publishing upside down." The missing piece was the software. I knew pretty much all the stuff that was going on, and the only package that would actually take advantage of that was PageMaker, which is why I told Apple you need to talk to Aldus, and Aldus, you need to talk to Apple, because that was the key connection.

Grad: There was a synergy that you were providing.

Seybold: Exactly.

Richard Ying: In the 1970s, my job was to make sure that the people, the editors and reporters, would use the computer to do their work, to type their stories. Jonathan Seybold said that the difference is that when you type on a typewriter, you pull the paper out, you fold it up and put it in your pocket, and you own the story. You don't need the device with you that is necessary to read your story back. He said, "You have to give them enough other benefits for them to want to use computers." I think that all of the technologies that we have talked about were enough to tip over the hesitancy of relinquishing your work, your artistic output into a machine that you can't rip out and put in your pocket.

The Engelbart Demo and Scaling and Cost Factors

Grad: That's a hell of a list. One of the things I wanted to have on record here was exactly this. Some other things may come up in our later discussions, but this is a framework of all the pieces, and it is amazing. It took longer to realize the potential than I would have thought because you didn't have your first successful businesses on the software side for the general market, other than newspapers, until the mid-1980s.

Ying: It was too expensive until then.

Grad: You were thinking the hardware was the key issue. Interesting. Now who wants to tell me about the Engelbart demo: Bob Sproull, do you want to start?

Sproull: I can't do justice to this. You need to talk to Alan Kay.

Lampson: We don't have time for that.

Sproull: We don't, but it's all on video that the Computer Museum owns. I really don't think we should go over it. It didn't have a lot to do with publishing, per se. Of the themes we've talked about, it was much more about hypertext and communication among people and especially enabling teams to work together. By the way, that is another thing we haven't said much about: whether there's a publishing team in a newspaper or a team putting together technical documentation. The demo used the mouse, and it had interactive displays. They were not all that pretty, nothing was in those days, but it was a tour de force compared to what most people in the audience were able to use in the way of computers at the time in terms of interaction, what it was being used for, and the potential. I think everybody seeing that demo imagined what computers would be like. There's one thing that it didn't dwell on perhaps as much as it should have, and perhaps this was an uncharacteristic Engelbart modesty: he didn't say that the way the technology's going, this is all going to be much cheaper. Even if he didn't predict Moore's law or tell everybody that we were well on the way, perhaps he could have imagined that the reason it was going to be very important is it was all going to get cheap.

Lampson: That's certainly what we thought at PARC. I mean, we used to say that we were building time machines. We were building things that were going to be commercially viable 10 years in the future, and we were building them today because there was an enormous amount to learn about how to write the software and design the user interfaces and all those things that are relatively independent of the cost of the hardware. That was brought to mind because what Bob said was absolutely true: that the two things that Doug Engelbart thought were most important about the online system were the hypertext-like links and the teamwork. Neither one of those things played a significant role at PARC.

John Markoff: Just two footnotes. Bill English had to invent a display system to make the Online System (NLS) visible to the thousand best computer scientists of the time, and Bob Taylor knew that that had to happen and underwrote the invention of that display system to make it possible for people to see it. Also, Engelbart should get credit for understanding the importance of scaling and cost. He gave a paper at the 1960 conference where he essentially laid out Moore's law before Moore did.

Geschke: One thing that hasn't been mentioned that also occurred at PARC at this time was the iconic display for how you work with the interface on a computer that affected the early precursor of Windows, and all of that was done as part of the Small Talk Group. While we don't necessarily factor it in because Small Talk didn't become part of it, it was really what motivated Jobs and other people to understand what the display interface could look like.

Brainerd: There was one other development that I saw at the Walker Art Center in this timeframe. The University of Illinois' Plato went through iterations over the years, and I saw Plato 4, which was a 512×512 matrix plasma display with 16 infrared (IR) points with a stylus. It had a lot of different elements, but it approached the problem from a different point of view. Then CDC took it over and eventually it died a slow death sometime in the 1980s. I mention that because I think there's more to the visual interface story than just Xerox PARC.

Grad: What you're telling me is that the Engelbart demo motivated people, gave them a vision that wasn't there before. Is that essentially what you're saying? Was it critical to the mission of Xerox PARC? The question is really, to what extent did the Engelbart demo affect the direction that Xerox took or that other people took? Did it motivate them or simply give them some tools they could use?

Lampson: It didn't give them tools. It was motivating.

Sproull: Some of the people from Engelbart's group came to work at PARC. They were not terribly central in what went on, but they were there. We also owned one of the workstation machines that had NLS running on it. It was sitting in one of the corner offices, never used, but it had a keyset and a mouse sitting there to remind us of what NLS was. I think the NLS ideas were certainly discussed. There were many people at the lab who were familiar with NLS and its demo. Don't forget, they were just across the Valley. There was a fair amount of traffic back and forth.

Lampson: Most of the Engelbart team was working at PARC.

Grad: What's fascinating, again, is that a lot of the work ended up here in this area, physically close. It appears that a lot of the software development was just national software

products that had no focus; people were all over the country. Here, this became a nexus for this whole area of development.

Bigelow: One comment about the Engelbart demo. I was a student and teaching assistant in typography at the San Francisco Art Institute at the time. I didn't go to the demo. I hadn't heard it, but when I did hear it, it was motivational. I remember what I said: "Far out, man. That's groovy." It was the 1960s, remember. <laughter>

Warnock: Bob Taylor was head of ARPA, and there were 20 ARPA sites. The best computer scientists in the world were at those sites. He came to Utah for about a year, and then he took the job at Xerox to form the lab at Xerox PARC. He cherry-picked every good student out of the whole ARPA community. That's why the concentration of software was at Xerox PARC. He literally cherry-picked the best graduate students from all the best universities from all over the country: Carnegie Mellon, MIT, Harvard.

Geschke: I don't think there was an engineer in this field who wasn't an ARPA product that I remember.

Walden: Yes, Taylor also cherry-picked managers and employees from other companies like Ron Ryder.

Grad: Jonathan shows a whole path going through the newspaper world. That was not the Xerox PARC path, but the Atex comes from that.

User Interface Development

Seybold: I'm separating the platform from the application software. The platform was the path that came through PARC. The application software for desktop publishing was the path that came through newspapers. There were two parallel paths. There's another piece to this PARC path that I think is really, really important and we haven't talked about: the user interface. People seem to still have the concept that you can dream up the user interface out of your head, you can implement it, and it's going to work. It doesn't work that way, okay. Steve Jobs got people to think that one person can do it, but in most of the world that's not true. I'd like one of you to talk about how the object-oriented system came about and what was done in terms of user testing. How did we get to the current user interface metaphor that we still have?

Lampson: I think they just thought it up out of their heads. It's actually a very interesting story, because when we did Bravo, we made a deliberate decision that we weren't going to work on the user interface. Not because we didn't think it was important, but because we thought it was hard and we didn't have the resources. So, we kind of hacked together a user interface (UI). The only thing we did that was careful was we isolated it into one module. We were very

fortunate that Larry Tesler and Tim Mott came along. They needed an engine, and they wanted to work on UI, so they tore out our module and put their own stuff in.

Grad: This is exactly where we want to go. Jonathan uses the word “platform.” Do you view the work at Xerox PARC as providing a platform or as providing the software?

Warnock: A very universal platform. The software is the platform.

Grad: No, it's not.

Geschke: Yes, it is.

Grad: Tell me why.

Geschke: If you require that everyone think of themselves as programming from a naked piece of hardware, and you're going to build everything from the ground up, good luck. Not many people know how to do that. But if you give them layers of abstraction that they can work on top of, then you have the beginnings of a real industry.

Lampson: Most of the manpower part went into hardware, not into software.

Grad: Chuck talks about the importance of PostScript. Jonathan talks about the importance of PageMaker. We talk about FrameMaker. These were the programs that actually ended up running and selling and used the platform. I always thought of them as the things that were underneath that.

Geschke: Paul did not build the user interface on top of which he implemented PageMaker. That was part of the Apple world.

Grad: You're backing Jonathan's statement that the software has a different path, and it comes by way of a different route than the platform on which all that software is built.

Lorenzen: I think we're defining platform in different ways here. There's a hardware platform; that's one thing. Then at the software platform, you have the operating system layer, and the graphical user interface ends up in the operating system layer. Then anyone—like Paul writing PageMaker, us writing Ventura, or Steve writing FrameMaker, depending upon what operating system you're writing on top of—just has to call in the functions that are already there. The graphical user interface is part of the platform, but it is software.

Grad: I absolutely agree. All the applications that I built were built on these platforms that all the guys up in IBM Poughkeepsie and the other places had constructed. The question is, was the primary work then at Xerox PARC to build the high-level systems software platform on which the applications could be built?

Lorenzen: It influenced all the other platforms that came later. Everything that Apple did, the Apple OS, Windows OS—all of those things were just clones of what came out of Xerox PARC.

Lampson: That is a gross oversimplification because you can't engineer the platform properly if you don't have the apps. We didn't have an app like PageMaker, but we had a lot of other apps and there was constant interplay back and forth between the fundamentals provided by the OS and the graphics, the fonts, and all that and the demands of what the application writers thought they needed and what the users had to say about what the applications were good for or not good for. This is why, for example, we didn't do spreadsheets at PARC, because we had no customers for them.

Knuth: I'm not sure I look at this correctly, but in a way, in the 1970s, if you tried to make a rank-ordered list of the top 100 software people in the world, the people who were able to make computers do the most tricks in the best way, you would find that there would have been maybe five out of the top 10 at Xerox PARC. Out of the next 10, there would have been seven of them at Xerox PARC. Out of the next 10, 9; after that, 10, 10, 10, 10. It was incredibly cream of the crop. As you said, Bob Taylor built it. Now on the other hand, when you have such brilliant people working, they aren't necessarily the best people who are going to make a program that's usable by your cousins.

Ruckert: Mere mortals.

Knuth: That takes a completely different kind of skill.

Grad: Good point.

Tool Development and Experimentation at Xerox PARC

Sproull: Allow me a few minutes to say a few things about what we actually did at PARC around the imaging area. Despite your comment, I don't believe much has been written about the details. You know about the input-output and the printing, but not perhaps much about the decisions that were made. As Butler said in his introduction this morning, we weren't building publishing systems. We weren't even really building imaging systems. We were trying to do experiments on organizational office information systems, and it was clear that graphics was going to play a role in that. When I got to PARC in Christmas 1972, Gary Starkweather had the first laser scanner of the second printer going and Ron Ryder was building a character generator design that Butler, I think, was heavily behind in order to drive that at 500 bits per inch of page a second. That was a fairly tough row at the time to hoe. It was controlled by a Data General Nova sitting on the network or what passed the as MCA network, if you can remember that. As Butler said, it was clearly going to have to be a server.

This for me was important in terms of what we had to be working on. There were going to be printers of various different kinds, different sizes, different speeds. It clearly could do only what the character generator could generate on that printer. We couldn't do graphics. There were some elaborate hacks that I won't bore you with, but they were hacks to generate things that looked like graphics by creating very special-purpose fonts. William Newman and both of us with backgrounds in interactive graphics, device-independent software packages for line drawing displays, and so on put together this design called Press, which was device independent. It was an attempt to cater to text, line graphics, area graphics, and halftones. We thought we could convert the whole thing to print on a raster display. When you talk about platform, by golly, that was a piece of the platform. Several people, myself included, went off and started working primarily on software to print that format at various different speeds, different color, different precisions, what have you. Other people went off and said, "You know, there's other pieces of this ecosystem that have to be put together like fonts." We had Patrick Baudelaire and our spline expert Bob Flagel, so we chose to use regular splines, not Bezier curves. Even the Utah guys had been experimenting with them, but to us, the fact that they had a superior user interface property had not yet been demonstrated. We made the wrong choice there, but at least we chose cubic outlines. I remember Don, I actually came and talked to you about that choice. You tried to talk me out of it in favor of your more stroke-oriented scheme. Frankly, we knew how to do scan conversion. We knew how to do scan conversion from curves and actually do it without glitches, and we were interested in getting on with the printing experiments, so we choose to use outlines.

As Butler was saying, there were apps, simple ones. Don't forget, we only had a 64K, 16-bit machine when we started. These were things to make line drawings. These were things to do bitmap drawings by hand. Think of them as early paint programs. They were in black and white, markup. The draw was the line drawing program done by Patrick Baudelaire. We experimented with color. One of Liz's associates at XEOS, Dale Green, had the best Xerographic machine on

the planet at the time. He not only understood the Xerography, but he captured all the great scanning resources as well. He had a 768-bit-per-inch four-color machine.

As William and Newman and I were doing the second edition of our book, we decided to use all of these tools as a sort of stress test. We used Pub from Larry, which had been modified to produce Press files as text output. We used markup for some of the bitmap stuff. We used DRAW for all of the line drawings. There's not a single line drawing in the book that is not done with DRAW. Many thanks to Haishan Sargent who did almost all the work. We wrote programs to combine it all and do the page composition, etc. and then printed them on Dale's machine all in black and white at as high a resolution as we could on hard surface paper. Despite what you said this morning, they simply made films from those pages directly.

Grad: Yes, you had this really cool printer. <laughs>

Sproull: We did have a cool printer, but we were not foolish enough to do halftones this way. The half toning that you did for a Xerox machine was quite different than what you did for an offset machine. There was no way we were going to make halftone dots on Dale's machine that had the right shape for an offset printer, so that was done with Exacto knives. We didn't do that. <laughter>

Sproull: That was the one thing we surrendered. The little piece of the desktop publishing story here that I think may be interesting is what we discovered in doing it this way with all of this sort of hodgepodge of tools. By the way, these were all experimental tools. None of them were products. They were still supported by their authors who were still alive and thankfully available to us to make changes, and we did a lot of hacking ourselves. But this was a book. Producing a book is not a newsletter. It is a database publishing activity, and what you have is what today would be called a tool chain that you run. After you make a change to some piece of the book, you re-execute the tool chain and the changes that you've made are propagated to the final result that you can then inspect before you actually print. So, it's more like the way software is engineered, modified, and released today than it is the way you put together a letter to your friend. End of story.

Grad: That's very interesting.

Alto, Dover, and Office Products

Lampson: I have one thing to add to that that some people probably know, but maybe some don't: Press, which Bob just referred to, was the prototype for PostScript. It wasn't that we didn't know that we needed a PostScript-like thing. In fact, we knew that very well, and we had one.

Knuth: I just want to ask Bob, I seem to remember one of the big mantras at PARC at that time was to be the paperless office.

Sproull: Well, this was always a source of some humor. Obviously, our bosses didn't want to go paperless. I mean, the Webster and eventually Stanford bosses didn't want us to go paperless.

Knuth: I enjoyed one day sitting next to the Dover at PARC watching all you guys printing out your email.

Sproull: Let me say one more thing about the Dover, because it was an example of teamwork at PARC. The Alto has about 30 megabits of memory bandwidth. Not bytes, 30 megabits. Printing a page a second on a 384-bit-per-inch printer takes 12 megabits per second. That's just sending the video to the printer, so you don't have much time to do other things. We never thought the Alto could print on a page a second machine, which is what the Dover turned out to be. We had a meeting one day—I can still remember this meeting. I can picture it, and I know who was sitting in which seat. I was sitting across from Butler, and Butler was facing the window. Those of you who've ever been in meetings with Butler know that he will sometimes get a sort of faraway look. He's looking just over your head, and there's something in his hand, a pen or a pencil, and he's scribbling and he's thinking really hard. Butler made a suggestion in that meeting about a simple way that we could build a single board that would plug into an Alto that would drive a page a second machine at 384 bits per inch, text and rules. Horizontal and vertical lines we could do. <laughs> Sarah Orenstein and I went off and did the hardware and software and made it work. That's what became the Dover. It was an otherwise vanilla Alto.

Shoch: Two quick things, if I may. Don, on your point about the printing volume and so on, one of the stories we used to tell to reassure the rest of the corporation was that we had the data from PARC on how the volume on the copiers went way down, but that the volume on the printers went up to more than offset it. Therefore, the total amount of paper that came in the loading dock was greater than it would have otherwise been. All those copier guys should relax, because we're going to have lots of ways to sell toner.

The other point I like to make, and these guys did even much more of it than I did, is that we have to be a little careful about, "the Xerox platform" or "the result from Xerox." It was a very iterative process over a number of years and the blessing that I took away from it is that the cycles were very quick. Because we were an industrial research center, we weren't dependent on annual research contracts from somebody else. If you didn't like something, you could go talk to somebody and have a conversation over a cup of coffee and fix it. So the Alto comes up as a text-oriented interface to the exec. That's all there is. Then somebody writes Neptune, which is a finder-like but not with icons. It has point and click to file names, but it has no icons. Then you get Smalltalk that has icons and the graphics include markup and SML (???) and

FRED and Icarus and printing formats that go from Press to Interpress. Did Jam ever get built, John?

Warnock: We printed from it, yes.

Schoch: I could barely reconstruct all of these pieces, except that was a furious pace of iteration. It's really hard to sort of say, "Oh, this did that." Somebody would see a feature in Gypsy, and it would show up somewhere else and then, "Oh, wait a minute." You look at the software catalogue, and every month there's a new thing. "Oh, this got released and this got fixed, and FRED just went from 7 bits to 8 bits in the coding and everything." There isn't a single thing that I think of as the platform. I think there's a paradigm that emerged. You get to the end with all of this. and it all looks easy. At the time, it was a free for all. <laughter>

Office Printing and Publishing

Grad: You're doing what you imagine might be the office of the future. Is that a fair statement? How many of you here were in Xerox PARC sometime during the 1970s? How many of you were working at Xerox PARC then?

Grad: Six. So many of you were there, and part of this process. Does publishing as such drive you at all as an application area? Or is that simply something that happened, or you saw happening? I don't think of an office system as focused on desktop publishing.

Unidentified Speaker: Publishing is absolutely part of the office system.

Geschke: That's part of what an office does.

Grad: But an office doesn't publish, it prints.

Crews: An office publishes all the time.

Grad: What does it publish?

Crews: It publishes its marketing materials. It publishes its annual reports. It publishes its presentations. It's a publisher.

Grad: That's a good point. and I hadn't thought of that.

Crews: Now the question is, what happened? How did I, as an individual, become a publisher? That's really the secret about desktop publishing. What motivated me as an individual to really want to be a publisher, to do my newsletters?

Grad: I think that will come up later with Paul. The market in the 1980s was a little different than what the envisioned market was, which we will get to later.

You made a very good point, Liz. We were using those big printers to run all the stuff, and you were looking for the next stage, and that wasn't it.

Warnock: The early adopters were horrible amateurs. They used every font on every page. Their design skills were...

Sproull: Negative.

Lampson: Negative.

<laughter>

Warnock: They didn't understand about fonts or topography or spacing or layout or any of those things, but they were enthusiastic. The people who did do those, like all the magazines at the time, said, "Voila. I can reduce my budget by a bajillion dollars by doing this myself."

Grad: Somewhere in there is a switchover from publishing happening in an office to making it something that you as an individual can do. That seems to be the massive change in the picture when we get to the 1980s.

Lampson: We needed all these organizational and structural apparatuses to enable the individuals.

Warnock: Early at Adobe, we wanted to make every person on Earth a master builder so they could do everything themselves.

Tesler: One thing to remember is that Xerox owned PARC and funded PARC, and Xerox was at one point around that time the second largest company by market cap in the U.S. and was in the top 10 for many years. If Xerox was going to be looking at the office of the future, this was going to be an office with printing. There was no doubt about it. When somebody from PARC walked into a client interested in running an experiment of some kind, the reaction was, "You're from Xerox, the printing company, basically." That was just a natural fit.

Ginn and Xerox PARC

Hansen Hsu: I want to hear Larry talk more about Ginn and how Ginn drove the development of Gypsy.

Tesler: I joined PARC in February 1973. What I only partially hid from everybody was that I was not so much interested in the office of the future, I was more interested in the print shop of the future. I tried to get myself on projects about that. There weren't any at that point, so I got roped into spending some of my time on the office of the future, and some on just general visioning of technology.

Then by surprise, just as I was at the point of maximum disappointment that this wasn't working out the way it should, my boss Bill English called me and said, "We've been contacted by Ginn & Company in Lexington, Massachusetts. Xerox acquired them a few years ago, and just like every other Xerox division, they're paying a little tax to the company to fund the research center, but they're not seeing anything being done there that would help them." He tried to explain how we have Alan Kay Education Software for their textbooks and so on, but they weren't buying it. They wanted to see products in the pipeline, and they didn't even care about the research part of it. They came to find out how they were going to get products that they could use in the preparation of their textbooks. I thought, "Wow. This is an opportunity," and Bill sent me and a few others to Lexington and spent a few days there studying what they did. They ended up hiring somebody named Tim Mott who became famous for the other thing later, but he got a job at Ginn as the person to work with PARC. He was a very good software engineer and very good usability person.

All of a sudden, there was an official project at PARC that was about publishing and the publishing industry, and the people involved in it grew. The usability people were hired: Tom Moran and Stu Carr had just gotten their Ph.D.s from Carnegie Mellon. Alan Newell, who was their advisor, became very interested in making sure that the products that were built out of these PARC technologies were usable, just to make a simple statement out of it. That answered some of the questions that have come up.

Font Design

Grad: The font design work was obviously significant. I think there was a tie-in with Mergenthaler at one point. Was that with Xerox PARC? Did you guys work with somebody else?

Crews: When the Xerox Corporate Font Center was first founded to help develop fonts for the 9700 printer and the electronic printers of that time, we were developing 300 dpi fonts, literally bitmaps. The actual development staff there was very taxed because it was primarily being used for the insurance industry, and the boilerplate was then added to specific data that

had to be put in. We would get requests to bastardize the Universe font into a fixed-pitch font. We made 1,000 versions of them because somebody had designed a form that wanted literally this many character per inch and that many characters per inch, and it was overbearing because we just kept regurgitating the same thing and bastardizing Universe and Helvetica until we got the fixed-pitch fonts.

In addition to that, it was very important to have the real rights to Roman PS and Courier, so we negotiated the contracts for that. It occurred, I think, to us here that there's going to be more need for additional typefaces because the insurance industry wasn't the only one that was going to adopt electronic printers. We went off to Mergenthaler Linotype particularly and asked them if they would help us, literally, make 300 dpi fonts. We negotiated a contract with Mergenthaler Linotype to build and make 10 typeface families for Xerox.

Mergenthaler and ITC particularly were questioning how many point sizes we could build. Could we build a 9-point, 11-point, 12-point? I'll never forget Ed Rondthaler at ITC literally taking a typeface design and making a bitmap of it to show me what it would look like at 9 point. We decided on 10 type sizes, and we had Mergenthaler create those for us.

Geschke: This taught us the lesson that if we were going to do printing technologies as we had envisioned, we would have to hire a monastery of people to make bitmaps. That wasn't going to work, so we clearly had to apply a new technology to that.

Frankly at Xerox, we never got to that point where we actually had outline technology that was usable in a printing device. That's probably the key inventive thing that helped launch Adobe.

Warnock: When we started Adobe, we didn't know how to do it. We had no idea how to do it. What we did know is that we either solve this problem or we would go out of business.

Lampson: You understand what the issue is. The issue is that if you rasterize an outline font in a stupid way at a low resolution like 300 dpi, the result looks crappy.

Geschke: And the stems will be different.

If you do the letter M, the middle stem will be thick and the two side stems will be thin.

Lampson: Right.

Sproull: I was just going to say that our PARC fonts were just there to get barely legible characters on the page. Every printer we had distorted those one way or another and compensating for zero graphic effects, which essentially is one of the device-dependence issues. It was a key thing that we did not attack at all.

Seybold: To compound the problem, the screen resolution was less than the printer resolution.

Sproull: Absolutely.

Geschke: Absolutely.

Seybold: And so, you end up with this anomaly. You have to look at the hard copy on the screen, and the type goes like this. It's impossible to read. When you try to make the screen match up with the printer, it looks terrible. And if you make the printer match up with the screen, it looks terrible.

Sproull: Yes, sir.

Brainerd: And that's worse.

<laughter>

Crews: The other thing I'd add that we discovered is that when you start working with typographers and type designers, you realize very quickly that typography requires additional character sets that were not available on the regular keyboard. It was really imperative that we literally work with a standardization committee so we could standardize where the ligatures were going to be stored and all the other additional characters—the copyright symbols, registered mark symbols, etc. Then we addressed the issue of foreign languages and accident characters. There was an enormous amount of effort that had to be put in as we were developing the time phases so that we could make sure we had the designs that would allow people to be able to print in multiple languages.

Grad: But you were still stuck with the bitmaps you're telling us.

Lampson: Yes.

Crews: That's right.

Brainerd: When we get to the 1980s, we'll tell you the fascinating story.

Warnock: How that was fixed.

Grad: We'll get there. Let's stay with the 1970s.

Brainerd: I just want to make sure before we finish the 1970s that Jonathan and I have a chance to talk about the newspaper publishing industry, to make sure we're going to cover that.

Grad: We may have to give up on the 1980s, but we will finish the 1970s. <laughs>

Brainerd: There's no question that Xerox and PARC and all that framed the technology and the vision.

Seybold: Can we start the newspaper now?

Optical Printing Technology

Grad: No, I've got three more questions and then we'll do it. Font design, the laser printer—you all said that was critical to this work here.

Sproull: Gray Starkweather was an optical engineer who worked for Xerox and Webster. Don't forget, there was a lot of optics in copiers. He made a key optics invention that allowed a scanline to be a uniform width across a wide span from a fairly narrow throw. If you wanted to make a throw 10 feet long, sure, no problem, but you'd have mirrors and mirrors and mirrors and it would never fit in your copier. There were a bunch of systems issues too. I said the scanline was at a page a second—you can do the math, but they were going pretty fast. You have to synchronize carefully the video you're sending. You have to be able to detect when that scanline is starting to cross the page and when it ends. I think there were eight-faceted polygons, so it's going pretty fast. It's not supersonic, so you're not driven out of the room, but very nearly. They made a big screech early on, and there was a bit of wobble. There can be some wobble in the rotor, so you have to synchronize, and you have to detect when the scanline passes. When it ends, you build a phase lock loop to generate the video signal and be able to compensate for the various mechanical aberrations that the rotor introduces.

Over time, the aberrations got better as people designed better and better rotors and so on, and the technology improved substantially. You can tell early Xerox PARC documents because the scanlines went vertical on them and they went to the 11-inch direction. You can see a little bit of drifting up and down as you go across the page. That's because we hadn't quite got the synchronization circuits to work well with the scanner. You didn't want it to go haywire if it missed one scanline passing. Gary and Ron Ryder by and large did the circuit and system stuff to make all that work. With slower copy machines, it was easier. Ultimately, that's the technology that was put in the Xerox 9700. This is what Liz (Bond) Crews was doing, all those 300 bits per inch fonts at two pages a second. That was the flagship Xerox printing product that made all this worthwhile, as far as the company was concerned.

WYSIWYG Capabilities

Grad: You did the stuff on WYSIWYG there?

Lampson: Yes, the result was typesetting that any typographer would consider to be unbelievably crappy. But it was good enough for us. It was a lot better than what you could do with a dot matrix printer.

For that matter, the XGP. Our goal was to make it so that you could print at the very least a beautiful office memo. That was the main focus of WYSIWYG. You could look on the screen and see something where you had the beautiful Xerox logo and the beautiful Xerox logo font; it was positioned in the right place and you could have other stuff surrounding it, various different fonts and so on. What you saw on the screen, to the untutored eye of the computer engineer rather than the professional publisher, looked just like the printed page.

Grad: It looked good enough.

Lampson: It was close enough for our purposes.

Sproull: For example, the widows were in the same place. The page breaks were in the same place. The indentations were roughly the same.

Lampson: Yes, the line started and ended in the right place.

Sproull: For most office documents, it was fine.

Knuth: I don't remember who told me this, it might have been Charlie Ying, —there was a question about taking this WYSIWYG type of thing to the people at newspapers who set the classified ads, because the many pages of newspaper classified ads were a huge keyboarding expense. Basically, the people at the newspaper said, "I don't need this. I know what it's going to look like. You know, I know this line's going to be italic and this centered." This was a case where the ordinary user can't separate the visual appearance from the content, so that's why WYSIWYG was revolutionary, but not for the professionals, for the ordinary user.

Canon

Brock: I want to ask a quick question about laser printing at Xerox and its patent protection, Did Canon quickly or at least relatively quickly within a matter of years get into this?

Lampson: Ten years later.

Sproull: Yes, not that quickly.

Brock: Was it that long?

Sproull: 1980s, yes.

Lampson: It was about 10 years.

Brock: What was the patent protection like? Was it licensed by Xerox?

Seybold: Let's be clear. Canon made the copier engine. It was Hewlett-Packard and Apple-Adobe who turned it into an imaging machine.

Seybold: All of the output part, all of the laser output was not by Canon.

Brock: I see. Essentially the laser part of it was the work of Hewlett-Packard and Apple.

Sproull: I disagree.

Bigelow: Canon made the laser printing engine, and all of the software architecture above it was from Hewlett-Packard, Adobe, and Imagen. The engine was Canon, and somehow Canon got past the Xerox patents. One method may have been by writing black instead of white.

Geschke: They'd be writing horizontal instead of vertical on that.

Lampson: I don't think there's anyone in this room that actually knows the facts about this, so we shouldn't speculate. We'd just be making it up.

Knuth: We'd try to tell them what they should do in order to drive it. After two years of frustration, Canon decided not to and that's why Imagen was founded.

Bravo and Gypsy

Sproull: Talk a little bit about the modeless stuff. You haven't talked much about that.

Tesler: First of all, Gypsy was technically Bravo with the user interface module removed and replaced by a different one. Instead of there being two loops, one for capturing characters that are being inserted, the other one for looking at commands that are issued after a suitable termination character, Gypsy only needed one event loop because you were never in a

character mode or a command mode, you were always in the same one mode. When you have one mode, you have none.

Sproull: The Gypsy innovation, I believe, was at PARC. Especially in the context of publishing with Ginn, it was the first appearance of a modeless user interface.

Tesler: Right.

Sproull: And that was what you were doing.

Tesler: More specifically, the use of cut, copy, and paste.

Sproull: Yes.

Tesler: We had a lot of skepticism, even from me, whenever we started working in the early days on cut, copy, and paste because it was not a common term in the general population. You had to be in a certain industry to feel comfortable talking about cut and paste. That all changed to the point where now cut and paste is a much more common term than any of the other terms that were used. We were afraid that ordinary people would consider it way too technical. We were going for a system that people could sit down and start using without consulting a manual or taking lessons. One thing that was obvious to anybody who had been using computers since interaction became possible in the 1950s was that there were a lot of modes in application software.

Grad: What is a mode?

Tesler: A mode is a state that the program gets into where the behavior of what happens depends on something other than what the user operated—say the key that the user typed. In all text editors at that point, if you wanted to search, for example, you would type a letter like "F" for find. You then type in string of things that you want to match, query. As you are typing what to search for, you can again type the letter "F," but the "F" means something different than it meant the first time. The first "F" meant find, but now the "F" is one of the content characters that you're searching for.

People would forget what mode they were in. They would not hit the key hard enough. They would hit the escape key prematurely and they'd get themselves in deep trouble by being in the wrong mode. I used to say, "Everybody complains about getting stuck in a mode, but nobody seems to be doing anything about it." Sometime around 1968, I decided I've had enough and I was going to start trying to find ways to remove modes from user interfaces, starting with text editing.

In 1968, I was not at the conference where Engelbart demoed NLS, but about a week or two later, a consulting client of mine told me that he had heard about it. He'd missed it also, but he'd arranged to get a demo to us at SRI that very day. A couple of people showed me a video and answered my questions. I went away like everybody else: "Wow. Everything's in here, but it's too complicated." The main problem was every keystroke put you into a different mode. Other than typing a find string or something, there were modes you went into and out of, but not on every character. Everywhere else in the user interface, every single character put you in a different mode and you didn't know what it was.

Schoch: I can give you one quick example of this problem. Prior to using Bravo, you would start at the editor on the Nova by typing the word "edit." At various points in my life, I would forget where I was in Bravo. I would be in command mode, but I'd sort of be working on something and I'd just instinctively type "edit," but I was in command mode of Bravo. I wasn't in the exec, so the "E" would select everything, the "D" would delete everything, the "I" would put you in insert mode, and the "T" would put a "t" in. You'd be sitting there and hours of work had just evaporated and your screen was blank. It's like, "What the hell just happened?" This is the victim of being stuck in a mode where your mind isn't where your fingers are.

Press/Interpress and the Launch of Adobe

Grad: Press and Interpress, how significant are they in this process?

Lampson: Totally.

Geschke: Very. Right at the heart of it. As Bob has already very clearly explained, the goals with Press were essentially identical to what we were hoping to do eventually with PostScript and before that with Interpress. The Interpress project started because we got a phone call from the folks working on the Star workstation. They said, "This is embarrassing, but we haven't been able to figure out how to get to the printer from the workstation, so we've got this document on the screen and we can't figure out how to print it. Can you help us?" I put together a group with John and Butler, Bob.

Warnock: And I was involved.

Unidentified Speaker: And Jerry Mendolsohn.

Geschke: Jerry Mendelsohn was down in El Segundo and Brian Reed. Only the two of us were physically in the same place; everybody else was on the Arpanet. We were emailing one another, and we never actually had an all together in-person meeting until the day we finished the project. When we finished it, we showed it to Xerox and they said, "Wow. That's fantastic. That's exactly what we need. That will be the Xerox standard for printing." I said, "Whoa," You

know, we'd only been a lab for like a year plus. I immediately got on an airplane and flew back to Connecticut to Xerox headquarters. I said, "I need a marketing budget because now we have this standard protocol and I need to get out and start talking about it." They said, "Whoa, whoa, wait a minute. At Xerox, we never talk about our new technologies and products until they're ready to ship." Everything we made had to be ready to ship.

I said, "That's strange, but okay. How long will it be before you ship?" "Well, at Xerox, it takes seven years to bring out a product." I said, "Seven years? That's two to three to maybe four generations in the industry you're going into. Somebody else will have done something. It doesn't matter whether it's any good or not. They'll have 100 percent of the market and we'll have none." And they said, "Sorry. That's the way it is." And that's what launched Adobe.

Office vs. Newspaper Systems and Users

Haigh: I've got a comment that I think will set this up nicely. We started with the suggestion that maybe the application is coming from the newspaper side and the platform is coming from another side. In terms of the capabilities of a computer being able to produce printed output, that's clearly true. But I think something that's come through in this discussion is that a big, crucial part of the platform is the WYSIWYG side, and that makes a difference for the kinds of people who can use it. As Don said, the people using the newspaper system didn't need it because they could see in their head what the page was going to look like from the tags. But remember, desktop publishing is about the individual who's not a typographer doing it from their desktop. One way to think about this, and this goes back to what you were saying about whether offices publish, is it's eliminating a kind of work; the individual can publish without needing to go to a print shop full of people who do that for a living; they could do it directly. What I would love to know is under the hood: how much of what's going on in terms of fonts, publishing, and so on is explicitly guided by the experience that's been built up in the newspaper systems, and how much of it is just a completely independent development?

Seybold: We actually have pictures to hand out that go to directly to your question. We didn't hand them out today, but we might want to now.

Grad: We do have wonderful color pictures that were done of the things that were going on in the newspaper publishing during the 1970s. You all can pick up copies on your way out. Study them in great depth this evening, and talk about them maybe a little bit tomorrow.

Seybold: I'll give you a hint. You will see PageMaker. You'll see the page outline. You'll see the text on the screen. You'll see PageMaker in those pictures. Just to set the background for what's happening parallel to PARC, we have this automation of newspapers happening during the whole 1970s. Until this time, computers had been used for support functions. As far as I can think, this is the first time in which computers were used as the core of a business. That is, the product of the business was produced on the newspaper. All of the interactions between

the people in that business, and these are not computer people, none of them are, take place on that system. It is the heart of the newspaper. It's made more difficult by the fact that at the time nobody in the newspaper knew how one was produced. The technology had been in place for two generations before anybody came to work for that newspaper, so you were trying to automate something in which everyone knows if I do my job at this point, somehow magically the paper appears. No one even knows how much wire service copy comes into the paper. No one knows how to quantify anything. And you've got to automate this, but all the users are nontechnical people.

Newspaper Environment in the 1970s

Seybold: Let's go back to the story earlier about typewriters being thrown out at *Portland Oregonian*. On the Monday after they first brought typewriters into the building, they were all found smashed at the bottom of the stairways. These were not people who were very eager to embrace technology.

The system has to be instantly responsive because you're on deadline, and it has to be failsafe. If you go down, you don't produce a newspaper that day. You can't lose copy. You can't go down. And you're going to be doing this with early 1970s computer technology, which Richard can talk about. This is a technical challenge. You're producing a newspaper that has newspaper text in it, but the text is laid out on a page and it's written and edited to fit that page, to fit the news hole on the page. You are managing news budgets. You are moving stories between different people and keeping track of who's made editing changes and who hasn't, so there's a whole management process here. You are capturing classified ads at the ad taker sitting with a headset taking the ad in, and as they're doing it, you are having to justify those things so that you can tell someone how many lines it is and therefore how much they're going to be charged for it and do the full charging and billing on it. And then you have a whole other section of people who produce display ads that are a very complex combination of text and graphics. And you've got people who lay the pages out, which means drawing boxes and knowing how things are going to go on the page and flowing the text copy into it. It is a complete system, like a complete package, under really difficult industrial strength constraints.

A big newspaper system would be 200 terminals. Response times of more than 2 seconds were totally unacceptable. On deadline, everybody's working. In fact, for the system that was installed at the *Washington Star* newspaper, we gamed that thing for election night because we knew that would be the heaviest load time. We measured it and we met those specs. The system had to be built for that.

In 1975, *U.S. News and World Report* was doing complete pages without pictures in place, transmitting them electronically to printing plants. In 1977, they were doing that with the pictures in place. So, from 1977 on, everything on the news page of *U.S. News and Report* was being

produced in Washington D.C. on their Atex system and transmitted to RCA Video Comps. This is, I think, III video comps, to be set in pages and on film that went right to plate.

Grad: What are the technologies that made that possible?

Lampson: Before we plunge into that, I'd just like to make one comment. The application that I can think of from the 1970s that is most similar to what you have just described is airline reservations. There are some obvious major differences, but there are a great many similarities and it would be very interesting to do a compare and contrast. Same requirements for fail tolerance. Same complete dependency of the business on it. All those things are the same. Of course, many of the graphics-related things are totally different.

Grad: I would argue with his comment about it being the first core system because IBM did build the things like airlines reservations. But let's stay with this on the technology.

Brainerd: So, I'm 25 years old, and I've just got my master's degree in journalism. I walk into the *Minneapolis Tribune*, and they give me an assignment to automate all of the prepress functions. They said, "You have a budget now for the next five years of \$1 million to \$3 million dollars to spend on technology that will automate this paper. Oh, by the way, there's the composing room up there with representatives of the International Typographic Union, which has 123 people. When you're finished, there'll be 21 of them left. We have to negotiate an agreement with them. Don't touch anything in the composing room. If you do, you may find your fingers bleeding because they're going to come down with a pica stick on it, a metal piece of <makes slamming sound> on your fingers." That was the environment.

As Jonathan said, this is what they said at the *Minneapolis Tribune* to me when I got there: "Every minute that we're late on the distribution is \$7,000 downstream. Every minute that you can't produce that paper, \$7,000 on your head, and you're going to be held accountable for it." I'm 25 years old walking into this.

Newspaper Industry Automation and Advances

Brainerd: Over the course of five years, we automated the entire newspaper and negotiated the agreement with the International Typographic Union. This had nothing to do with anything happening out in California. You guys had that world there. We had the practical thing of how to move these people and the technologies together. We formed a joint group of newspapers, called the Newspapers Systems Development Group, which few people even talk about anymore. In those days, I think they spent \$60 million with IBM, Raytheon, Singer, Omnitex, and names that you probably never heard of to write a functional specification for a fully automated newspaper system, which went nowhere. Because the problems with the labor unions and the people and the resistance to technology were so great in this industry and the

requirements technically were so great, we thought we'd get the best people in the world, IBM, to come up with a solution. They just kept writing and stalking us, but a full systems specification was written over a period of about three years, and that's when I came in in 1973.

That was going to be the future of the newspaper industry, independent of anything that was going on here on the West Coast. It didn't turn out that way, of course, from a technical point of view, but it did shift the thinking in the newspaper industry so we could work together and accomplish a common specification that wouldn't be unique to each newspaper. We could in fact work together as a technical team. Each newspaper had a member in this group. We could get vendors to actually produce a comprehensive system that in essence became the Atex system over time. It was an interesting time, for sure. As a 25-year-old, I learned a lot in a hurry. Independent of any of the stuff that you guys were doing, as you'll see in these pictures, we were just doing it differently. I think that leads well to Richard and Jonathan as to how we went about and created those systems.

Ying: I was an old guy when I started in the newspaper industry. I was probably 24 or 25. Actually, I'll go back a bit into the 1960s. I'd like to bring up a name that maybe we should look into, because I don't know that much about them. I worked for Hendricks Electronics.

It's a Mormon outfit, and that's where the funding came from. There's a lot of publications they put out. Money didn't seem to be an object, and we were given a lot of money to try out all kinds of new things like that big disk I mentioned earlier. That was in their office. Also, we got to use new technology like the brand-new PDP-8. That was before the PDP-11. They had the first keyboard computer, I think. It's a vector screen for generating text, and the intention was actually to get them to the editors, although I don't remember now whether they actually accomplished that. They also did a whole bunch of other things that I wasn't involved in. I wanted to bring up the name so that if anybody remembered anything to add it into the history.

Birth of Atex

Ying: Come the 1970s, Atex started work first of all at *U.S. News and World Report* in the production department—four terminals there. We have been talking about the small parts of the technology that came together back then to bring up the then infant publishing industry. I think we should spend some time to look at the user side. Who were the users? Who were the companies and what were their needs? Why did they want to automate even?

As Paul pointed out, there was severe resistance to automation. It is not that they didn't see the future coming. One of our customers was the Government Printing Office. I went into their printing shop; it was a whole floor, say four times this total area, of line casting machines. The walls were battleship gray. Nice paint until you tried to scrape it. It was all lead. The guy told me that they had more lead on the walls than they had in the little picks that go into the crucibles. They had melted. They lost tons and tons of lead every year into the air in front of the linotype

operators. The Occupational Safety and Health Administration (OSHA) was being organized at that time, and they saw the writing on the wall that they wouldn't be able to have those around. Yet, the union said that we couldn't change, so we had to be creative. What do we do about it, changing human nature?

At the time, the computers were not quite able to do the job that we wanted to do. We were all trying to simulate the future, to fake it. We had the most advanced computer that we could afford, the PDP-11, when we started, and there was less than 1 megahertz. We knew that we could not give one computer to each terminal. In fact, my directive was to get one computer to service 30 editors. The total intelligence of the terminal was like a standard television. We got it to run in non-interlace, so you get 400 or 512 lines sending at 60 cycles, but that's it. Generating the characters, doing everything, was custom work.

Thankfully, I have a brother who is into hardware and did all of the tricks to make that work. How do you get a 1-megahertz computer with 64K word, 2-byte word addressable space to service all those people? The OS available at the time didn't have a clock in it, so you can't do time slice multiuser. We had to invent all of those from scratch, change the computer to make a multiuser, multitask OS. All of that was needed to get the job done at a price point where we could actually survive. And we did.

The important thing that I think we learned was that the job is not about the technology. If we can't give something to the user that is a lot more than a typewriter, we can't change the behavior. They will stay with the typewriter. They will stay with hard copy. They want the physical ownership of their paper, of the printed, typewritten page. I think we spent a lot of time trying to give them all of that, the typography and the justification, because we had a monospace terminal and all of the editing features.

I think that Jonathan pointed out that a lot of those people had no idea what the total workflow was. We couldn't tell them, when you're done, send this to your editor. We don't know what it was. How do you get your copy back to the editor? You see these pointy things? It's a little run lead on the bottom, a pin about five inches tall. It's called a spike. You put your copy in there and a boy picks it up and delivers it to you. So, we need a key called a spike. You hit spike, and depending on your sign on, it knows who gets your copy and how many copies to make. It goes to your editor. It goes to your proofreader and legal.

That's the editorial side. But you can't make pages without the rest of the stuff. There are two other departments: classified ads and display ads. Classified ad people don't care what it looks like on the screen. "Lovely. We don't know how to put WYSIWYG on the screen, but we can break down the lines and tell you how many lines it is." The business people in the newspaper are in ad sales. Their job is not to take the ad. Their job is to listen to what you want to do and sell you lines. If you have a three-line ad and you run one word over, they'll say, "Add a few more words to it." Before you know it, you have one word longer. <laughter>

You run a one-day ad. "Sir, if you run Monday, Wednesday, and Friday, then you get Sunday for free. It's included." That's what their job is, even today if you call up to place an ad. All of those things have to be in the computer because the costs have to be instantaneously updated so that he can sell. We can't justify a classified ad system based on helping them type faster. There's no money in it. We have to help them. Classified ads are the same thing. PageMaker, how do you make up pages? Pages have real estate value. These aren't just pages—front page, back page, inside page, page next to news. You don't run an ad for United Airlines next to an air crash story. All of those laws have to be in the system. Of course, my job is to say, "Yes, sir. We'd love to do that, sir," and try not to sweep that under the rug until we're actually able to do that. We have a system worked out with the *Boston Globe* where pages have real estate value. If you have a particular breaking news, do you want to start taking away ads? At what priority do you make news enlarge the news hole? If somebody wants to put in a big ad, you need to shrink the news hole. The things we put in are the things that the customer wants.

Grad: Richard, you built the technologies and the product. And you ended up dominating the newspaper publishing business entirely with Atex.

Ying: Yes.

Grad: When did your products come out?

Ying: In 1972 we started working with *U.S. News and World Report*. We got the first check from *U.S. News* on Thanksgiving Day. That was a Thursday.

Grad: When did you start building the product?

Ying: Just about then because before that we had no money.

Atex Development

Seybold: After I gave a speech at the National Computer Conference, Doug Drane, Richard's brother, and I sat down and talked for a long time about what they were doing and what they wanted to build. At the time, my father was consulting for *U.S. News and World Report* but had a serious problem because they were changing printers. The only way they could do this properly was to move the entire magazine to electronic output by that deadline. I put my father and *U.S. News and World Report* together with Atex and said, "Of all the people I've talked to, these people look the most promising for doing what you're doing, but they're a startup company. They haven't built anything yet."

As Richard said earlier, they went to see Atex and eventually they wrote a contract with Atex that had milestones that had to be met and payments that had to be met along the way to make

this thing happen. Miracles of miracles, it happened. I got it done, and the magazine got produced.

Grad: We've been talking about all the technologies that were being worked on at Xerox PARC. You made the point that the things that led into what became this really came out of the newspaper work. What specific technologies was Richard working on?

Seybold: They started out just doing the text and the cut and pastes into the pages for newspapers. But what they ended up with by the end of the decade was a layout terminal where you have a graphic of the page and you have all this news copy coming in from these systems that has been prejustified and written and edited to the number of lines it has to be. You've laid a page out and you flow those news stories into that page. You have display ads that were produced on display ad terminals that have graphics. You put them on the page.

Grad: They built all this software from scratch, without any other foundation?

Seybold: Yes. They also built the hardware because all the software required their hardware.

Grad: How big a company are we talking about here?

Ying: At the end of 1970s, we were 1,000 strong.

Kirsch: When you guys met, how many employees did you have?

Ying: Maybe three of us.

Grad: That's what I figured.

Ying: <laughs>

Kirsch: They were promising to handle *U.S. News and World Report*, three guys in a startup.

Grad: It blows my mind.

Seybold: But there was no other choice. It's pretty amazing, isn't it?

Kirsch: How fast did you have to deliver?

Ying: In July 1972, we delivered the working product, and the first issue was July 4, 1973. In that time, we had to not just write the software and build the hardware, but to turn a plain vanilla PDP-11 into a multitasking, multiuser system with no display screen. You're talking about a TV.

Kirsch: No way you can do it that that fast.

Seybold: But they did it.

Kirsch: You weren't experienced, because otherwise you never would have thought you could do it.

Technological Foundations of Atex

Brainerd: They convinced seven of the major metropolitan newspapers. The \$60 million in 1973 that was invested in that specification with IBM convinced us it was possible. We became convinced they couldn't implement it on an IBM 360; it just was impossible so we needed some additional technology. That gave us the confidence and the understanding of the problem to allow us to take something like Richard and Charles and Doug's idea, converting a PDP-11 and putting some custom cards in it. Did anyone else do custom cards for PDP-11s?

Kirsch: Dozens of people did.

Brainerd: That's how we got there in that timeframe. We took advantage of what digital would give us.

Ying: The PDP-11 had this great thing called Unibus, one of the first chassis with a standardized, bus so we could do all kinds of things with that.

Grad: In the late 1970s, Mergenthaler was installing systems in large newspapers. Jonathan has told me, of course, they never were the major player in the business.

Seybold: No, the major players were Atex in the large newspapers. There were 24 companies making newspaper systems for newspapers large and small by the end of the 1970s. Atex was the largest company but there were 24 of them, mostly in the U.S. and a couple in Europe. Bear in mind the technological center for all of this was the Boston, New England area. The biggest concentration for all the commercial types of this stuff, was not the West Coast; it was the East Coast. This was a world apart both in terms of what they were doing and also geographically from what was happening here.

Grad: From the technological standpoint, which of the products we're going to end up talking about during the 1980s discussion are directly dependent upon the work that was done at Atex?

Unidentified Speaker: PageMaker.

Brainerd: The display ad layout terminal was called the GT68. I still have the spec; it's 24 pages. Basically, it was the next generation of microcomputer at the end of the 1970s. The GT68 was the 68000 Motorola, so it had the address space that we needed to do the bitmap graphic and we had that operating when Eastman Kodak purchased the company.

Grad: PostScript does not come from the work here?

Seybold: No.

Grad: That is a completely separate genesis, correct?

Ying: Yes.

Seybold: That's right.

Ying: Atex is a commercialization of all the products. We borrowed IBM's composition software and we added our contribution to it, and I believe that other people borrowed our contribution. A lot of the kerning, for example, came from the DEC.

Seybold: The hyphenation dictionary came from my father, who had licensed it to *U.S. News*, which licensed it to you.

Ying: Yes, we borrowed the fundamental technologies that you guys have done, and IBM and Digital Equipment had a TYPESET-10.

Grad: We're going to see in the 1980s two of the major pieces in the software world: PageMaker and PostScript. We then get FrameMaker, and we get other things that add on. We get the Ventura work that speeds up a lot of the process. Those are the fundamental things, plus all the other stuff we've talked about, the laser printer and the other things.

Seybold: The interesting thing was that we start out in the 1980s with this being personal publishing. By the end of the 1980s, the desktop publishing has subsumed these larger commercial systems, and people increasingly are basically building large, professional publishing systems around desktop publishing.

Brainerd: One other quick comment is about the customers: the reason they could do this in a year is the customers drove it. I was a customer. I was buying an Atex system, but they didn't have a functional specification for their system. I got on an airplane with one copy editor, and we wrote the functional specifications. The newspaper wrote the design document for Atex.

Ying: That's the answer for you: How did we get the product done in such a short time? We skipped writing the functional spec. <laughter>

When a customer wants it, we say, "Well, write it."

Brainerd: They told us to write it. We had writers. We got on airplanes every week, and I flew to Boston from Minneapolis and we worked for 48–72 hours straight, writing a functional spec that turned out to be 256 pages of what the Atex system did. We defined all these things.

Lampson: What happened to the \$60 million IBM spec that you told us about earlier?

Brainerd: The trashcan. But it was important because it convinced the industry we could do it.

Markoff: I have a short footnote on these editing systems that we've been discussing. My first one was not Richard's. I grew up on an SI system that was built by one of his competitors on the West Coast. From the point of view of a user, I just wanted to highlight Larry's point. The most powerful feature on the system that I had was that the top line of display was devoted to messaging. It was basically Twitter before Twitter. And the most powerful part of it was its simplicity. You were limited to seven lines that you could see, but basically, everybody just used the first line and it was such an important collaboration tool. Its simplicity was its usability and its power, and it's what made the system actually really effective for workplace collaboration.

Seybold: What Paul was talking is really crucial here. Whether you were a tiny newspaper or a large newspaper—they bought different kinds of systems—the buying process was always the same. A committee was formed on the newspaper that was dominated by the editors but also included the display ad people and the classified ad people and maybe a businessperson; it was dominated by the users. They would select a system by going around once these things were off the ground and trying them out. It had to produce the newspaper, it had to meet the specs, and it had to be a good price, but because the people driving this were the users, how easy it was to use and how natural it felt to them was absolutely crucial in selling the product.

At the same time this was happening, in offices, word processing systems were bought by bosses for secretaries to use. It was an entirely different mentality, an entirely different focus as far as user interface was concerned.

Haigh: We've heard the key thing here really is workflow. This isn't an individual system; this is about a whole bunch of people collaborating and it's all very much customized to what newspapers need. But people are going to hear this transcript and they're going to say, "We still don't have an idea of how page layout was actually done on it." We know with PageMaker that you pick something up and you move it around. I can reverse engineer from what you've said that there's a graphics display that is frequently updated when changes are made so you can see what a page will look like. How do you actually lay out pages? Are you putting a bunch of tags in? Is there any kind of pointing device involved?

Seybold: Go look at the pictures, and then ask me tomorrow.

END OF THE INTERVIEW