Timesharing/Professional Services Workshop: Session 11: The Demise and Legacy of TS-RPS

Moderator:
Burton Grad

Recorded: June 3, 2009
Mountain View, California

CHM Reference number: X5386.2009

Table of Contents

Timesharing/Remote Processing Services

Session 11: The Demise and Legacy of TS-RPS

Conducted by Software Industry SIG – Oral History Project

Abstract: Pioneers of the Timesharing and Remote Processing Services industries in the 1960s and 1970s discuss why it disappeared and what replaced it. They talk about the changes in technology, the evolution of business models, and the change in customers. They discuss the legacy of timesharing and its influence on email, e-commerce, networking, and now the ubiquitous individual access to computing services and data.

Participants

Name	Affiliation
Burt Grad	Moderator, SI SIG co-chair
Dick Bayles	National CSS
Frank Belvin	Interactive Data Corporation
Chris Brook	GE Information Services
Ann Hardy	Tymshare
Norm Hardy	Tymshare
Mike Humphries	Tymshare
Dick Orenstein	National CSS
Nick Rawlings	National CSS
Jeffery Stein	Online Business Systems
Mike Wyman	Interactive Data Corporation
Thomas Haigh	Historian, Univ. of Wisconsin
Chris McDonald	Historian, Princeton University
Doug Jerger	SI SIG member
Luanne Johnson	SI SIG co-chair

Burt Grad: I hate to name this session the "Demise of Timesharing," but unfortunately, it was really the truth, wasn't it? As a model, timesharing did disappear, and whether it has come back now, or not, is one of the things I'd like to have some fun talking about as we go on. But let's sort of pull it all together. You've mentioned, in a variety of discussions over the days, different reasons why you felt that timesharing was no longer an economically viable option. Technically, it would still do what it was supposed to do, but the economics were the issue. At least, that's what I heard. Go through the reasons. Minicomputers were certainly one of the things you said made a difference in the economics of the situation. Different prices on mainframes and Microcomputers: did those things make a difference or not? Were there other things that happened, different from those that were hardware driven, that you feel were significant?

Factors in the Demise of Timesharing

Nick Rawlings: IBM's VM [Virtual Machine] program was not available when we started. I mean, it was available free, "take it as it is," but it wasn't a viable operating system for a company that wasn't in our business. But certainly, by the early 1980s, it was. It was a system that you could bring in-house, and without too much fuss or muss, install it, and it would run reasonably well. So I think that was a software issue; it wasn't a hardware issue.

Grad: I'm going to clarify something I said yesterday about VM on DOS machines. It was used to run multiple DOS machines under VM. That was the way it was being used. I checked that out last night. So although it was used for timesharing on larger computers, it was really used as a virtual machine on smaller computers, rather than for its timesharing capability.

Rawlings: Well, as Dick Bayles mentioned, he thought that an early, at least official purpose for the [IBM] VM/CMS, which created a virtual machine environment, development was so they could test various operating systems. Mostly it was various versions of OS, MVT, and MFT, and whatever. But it also became a way of running multiple DOS's. And DOS had a much longer life because of VM/CMS.

Grad: In Europe it was just used everyplace, I was told. VM became a big selling thing. We announced VM in DPD [Data Processing Division], because the product divisions wouldn't announce it. So we went ahead and announced it in 1970. But the big thing is, it saved IBM's butt on timesharing on the big mainframe systems, because a lot of them got installed. All the customers who were going to get TSS ended up using VM as their operating system. So that's a factor, a different factor than we mentioned before. What were some of the other factors that came in? Norm, your thoughts?

Norm Hardy: Well, I think ultimately it was the PC that did timesharing in. Early on, I think there was a lot of motion toward the PC before the PC was really right. But it grew so rapidly beyond the 640K limit, that pretty soon if you wanted a program, an application that accessed a megabyte of RAM, you could afford it on your desk, whereas on the old mainframes, with the old pricing systems, that was an expensive proposition.

Grad: For example, you talked about your scientific and engineering applications. You still needed the power of bigger machines to do that kind of work at least until the 1990s, didn't you?

Norm Hardy: The early microprocessors were wimpy on floating point. The later ones exceeded the power of the older mainframes, so there are some dates to be re-examined. But even the heavy processing stuff nowadays could then be done.

Ann Hardy: And the SDS [Scientific Data Systems] 940 didn't have floating point.

Mike Wyman: The SDS 9300 had floating point, but not the 930 or the 940.

N. Hardy: Yes, we lost out to NCSS for any intensive floating point stuff. Even the DEC [Digital Equipment Corporation] PDP-10 was relatively weak on floating point.

Grad: Interesting. Okay, so we've mentioned that the micros made a difference, then, because they became powerful enough that they could run their own timesharing system. Because "it's my own system," correct?

N. Hardy: Yes.

The Internet Arrives

Jeffery Stein: I think there's a convergence of several things of why the industry changed. Back in the 1950s and 1960s -- particularly the 1960s -- there were very, very few people around that were technologists. There were very few information processing things that were done. I mean, there was payroll, and there was accounts receivable, and maybe inventory and scientific calculations, and all that. But very little was done. And then hardware came down in price. Plus, the biggest thing of all is that more and more younger people were coming into the field that understood technology. There was a huge shortage for years and years and years and years, of programmers, of analysts, of people that really understood IT. Also, there were many, many, many, many executives that were chicken. They didn't understand technology, and their way of taking care of it was something that they had to have, and they created the

data processing department, which was under someone in accounting. So we had all of these things going on.

Then the older executives left and we had more technologists coming in. You had more technologists coming into the workforce. And hardware prices kept coming down. More applications were being thought about that had to be done. So you had this convergence. Then the thing that really, really accelerated the change, of course, in the late 1990s, was the fact that we could connect all of these computers up. We had the Internet.

Grad: Was the Internet the dagger through the heart?

Chris Brook: Well you had to get the capacity you needed. If you had a mainframe and a PC, then that worked okay, but if you took the mainframe out of it, which is what we were offering, then you needed something to do the interconnection, and you needed to have a central location to do the various bits and pieces and then redistribute them. The Internet was an easy way to do that.

Grad: So you no longer needed your infrastructure, because the infrastructure was available to anybody?

Dick Bayles: When did the Internet really become available to anybody?

Wyman: 1995 or 1996.

Brook: That was mainly what it is. In the late 1990s the Internet came up, and so you could now interconnect two PCs, or PCs to a server, and share the exchange.

Grad: Let me ask my question again. Was that the final straw?

Wyman: Burt, there's a clarification, though, for what you're asking, because I think each of these guys that I've heard is right. They've explained how certain parts of the business went away. The last one, with the Internet, was EDI [Electronic Data Interchange]. None of us here had much EDI business, except for Tymshare. That was both an opportunity and a problem for us. Our first pure timesharing business, the departmental business, went away with minis. Departments would buy a mini. They didn't need to be heavily connected anywhere; they could do whatever they were doing locally. Tymnet and others like Tymnet (UCC, some of those others), they made it possible for people with their own computers that needed to be connected -- companies with many locations, or companies like I was talking about that wanted to have their customers access their applications -- that made that possible. So different things pulled away different parts of the business. But by the time you took away a few certain companies,

certainly Tymshare, you had had a critical mass removed, and you probably couldn't be in business any more.

Grad: Your points are well taken. Your view is that the Internet wasn't the killer of timesharing. Is that a correct statement?

Wyman: By the time the Internet came to be, we were already dead and buried.

Grad: Was that true? What was the final thing that killed timesharing? Was there a specific thing? Wyman says the minis.

Bayles: I think the PC was the end of it.

Wyman: Minis, PCs.

Dick Orenstein: Yes, minis and PCs, I think.

Rawlings: But with the PC, it was the applications, the software that was available on the PCs. The Lotus 1-2-3, the word processing. We were selling, on timesharing, word processing. We were selling spreadsheets. But there was no point in people paying even five bucks an hour to connect, if they could do it on their PC. And so I think the PC software killed timesharing.

Chris McDonald: Can I just ask a question? Were there applications that you had that were PC resistant, because they required a central database, or communication, or something like that?

Brook: EDI was probably the last one. It changed, and morphed to be B-to-B [business-to-business], but that was really EDI by a different name, with the clearing capability.

Grad: I'm not going to buy that. IDC [Interactive Data Corporation] had databases, and the databases had value.

Wyman: They're still in business and doing better than ever.

Grad: Wasn’t it timesharing that gave you the mechanism to have the access to the databases?

Brook: There are still databases all around. There are any number of places you can go to get information, whether it's the web, or whatever. You use the web now to get to those same databases you used to get to.

Grad: Yes, but it's the Internet with the web that lets you do it. I think Chris' question is a very interesting one. The database was your value added, and timesharing was your mechanism of access. Is that wrong?

Wyman: No, that's correct. Initially timesharing was the only way you could get to the data.

Grad: Prior to Internet, how else would you get to it?

Wyman: Through timesharing. We did bulk data delivery.

Grad: Okay, but my point is, the timesharing piece of your business wasn't killed until the Internet comes along to give us another way of getting at the databases. Am I saying it wrong?

Wyman: No, no, no.

Orenstein: The timesharing business, yes. The Internet didn't change his business at all. The mechanism to get to the business changed.

Wyman: The only thing the Internet did for us was to facilitate customers getting to us. They no longer had to have a modem and a phone line to dial up into a network.

Frank Belvin: But you also sold stuff, and sent data on floppy discs, by overnight mail. There were other techniques besides timesharing access through a modem that you provided at IDC.

Grad: When did IDC stop using the timesharing system? Or did they?

Wyman: They never did.

Bayles: I think the proper question is: when did people stop using it for just plain old timesharing? That's the question.

Grad: No, there are two levels. At the first level, you're right; it's a pure timesharing system. But at the second level, some of you added something onto it -- an application, a capability -- that kept the timesharing model alive for a period of time.

Bayles: It kept a communication mechanism involved in timesharing alive. It didn't keep timesharing alive.

Thomas Haigh: But remember, timesharing is an operating system technology, right?

Grad: Right.

Haigh: You're kind of blurring it here, to be the operating system technology and the business model that also includes remote communications. So to say they're getting access to it through timesharing -- they're not. They're getting access to it through a network.

Brook: Wait, why did you say timesharing is an operating system technology?

Haigh: Well, because it's a capability that, for example, is present in today's UNIX. In a sense, it's in Windows NT terminal server that lets you have multiple users with the illusion that they've got a machine to themselves.

Brook: UNIX doesn't have timesharing. UNIX is just an operating system for which you have to write applications that can mimic what we did with timesharing. Timesharing systems provided a whole bunch of automatic stuff that UNIX doesn't have unless you code it.

Haigh: Well, I think of timesharing on a core kind of level. What, say, would have been there, with the Dartmouth Timesharing System and BASIC and so on, is the ability to have a single computer servicing multiple users simultaneously, giving them the illusion that they have a machine at their own command. Right? I mean, they're literally sharing time. They're sharing the resources. There's only one processor there, there's only one memory, but the multiple users are sharing it. On a technical level, it's time-slicing.

Brook: But it still needs the applications. Whatever it is, people are going to timesharing to do stuff, right? Going to UNIX doesn't give you anything at all, unless somebody has written an application that runs on it. In which case all you've done is replaced your GE Mach III system with a Sun Server, or whatever it happens to be that you're running on. You've still got to have some reason to go there.

Haigh: Right. Anyway, the point of my remark was that we're reaching a period where we're blurring together the business model and the communications that get you access to the

computer. The operating system and application capabilities that would technically be called timesharing are becoming less useful. I've seen some discussion at cross purposes there, I think. We're just using the words in different ways.

Grad: That's a good point. Go ahead, Mike.

Wyman: Can I clarify, at least in respect to IDC? IDC, as it evolved from a pure timesharing play into an online information services play, continued to use a VM base for its infrastructure and all its applications. Whether the VM was used to support online timesharing for end users, versus running applications which produced the product which our end users made use of, was sort of irrelevant. And if you look at IDC's business today, I bet you they're still running the IBM 370, but there's very little online interactive access to the system except to access applications which provide online data, or online information.

Grad: So they're using it as an operating system.

Wyman: Exactly.

Grad: Which is the point Tom was making, in that sense.

Wyman: Yes, Tom's point exactly.

Grad: But are you saying you could use something else to do that online access, instead of using VM?

Wyman: They could probably, if they wanted to make the investment. They could use MVS. We were also running Solaris Systems.

Grad: Or anything else that has that kind of capability.

Wyman: Exactly.

Grad: Good point, though, that pure timesharing sort of ends at the end of the 1980s, or early 1990s. Where in time is the use of timesharing as timesharing, forgetting the applications or the database? Is there a point in time where that's no longer economically viable -- it's just not done? Is that the wrong question? Help me with the question.

The Migration of Applications Error! Bookmark not defined.

Brook: I think the whole model of timesharing back then was a dumb terminal (or marginally intelligent terminal) going to a very intelligent, very powerful mainframe with a lot of storage, etc. It was doing all the work for you, because you didn't have that capability on your desk. Once you could move all that stuff down to your desk, you don't need to go out there.

Grad: A client/server model has somewhat that same characteristic, though, doesn't it?

Brook: It does, yes, but it's more than that. For instance, let's take personal finance. I can sit here and have Quicken on my PC, and do 90 percent of the stuff here. And then go off to my bank computer, or my wealth manager, or whoever you're talking about, to get extra information I need, so that the guy down here can finish the job.

Grad: But if I need access to the corporate data, or to a lot of other source information, I don't have it sitting on my computer. Maybe today you do, but I guess I'm trying to get back in the early 1990s, before the Internet is around. Where do I get this stuff?

Haigh: You kind of threw me with that time period in the last comment. In a sense, the technology is still there. It's built into Windows now. You can create a remote desktop to a computer, and your PC basically becomes a graphics terminal and everything is running on a server. You can even use a remote desktop connected into a desktop PC now. It's a capability that's been built into Windows.

Grad: But you're talking legacy capabilities.

Haigh: The technical capability is still there. But it doesn't make sense anymore to do it just to get access to remote resources, because you've got so much computing power on your desktop. You do it now to get access to a particular environment. Or to an application that is running in a secure LAN [Local Area Network] somewhere. So you VPN [connect via Virtual Private Network] into it, and you run it on the desktop. It's more secure, and it's easier to maintain and so on.

Grad: I don't know if the others would agree with your reasoning. That's an interesting question. Ann?

A. Hardy: One simple way of saying it is that Grosch's law was repealed. Grosch's law, "the power of the computer is the square of its price," was repealed.

Grad: And that's gone.

A. Hardy: That's gone.

Grad: So is Grosch. <laughs> I guess maybe I'm seeing it too simplistically. That's what I was trying to say. All of you define yourselves, with the exception of Stein and to some extent GEIS [General Electric Information Services], as timesharing companies to start with. You started with that particular technology model. You started with those operating systems, whether it was on the [SDS] 940, whether it was on the [IBM 360/] 67, whether, as when [Ken] Ross was here earlier today, it was on a DEC machine. That's what you started with, and you looked for things that enabled people to use the computer at an attractive price. That's what you did. Then you started to add applications.

You run into a wall sometime at the end of the 1970s, or in the mid 1980s, as far as timesharing is concerned. Right? You all were saying that at that point it stops growing. You're doing everything you can to stem the tide. I heard some of you saying that you had to do a lot of things to stem the tide of erosion. It wasn't growing at the clip it had been; it was a different model. At least, that's what I heard.

Rawlings: Well it wasn't a wall. There was a slope.

Grad: Okay, "slope", fine. I'll accept "slope". Is there some point where there's a cliff? Maybe that's an extension of your terminology.

Rawlings: Well, I guess in the middle of the 1980s rather than the end of the 1970s.

Grad: So the middle of the 1980s is the cliff?

Rawlings: It's the point at which there are enough alternatives to timesharing.

Brook: End of the 1980s.

Grad: End of the 1980s?

Mike Humphries: By the time we go to zero, it is probably the end of the 1980s.

Rawlings: Where the alternatives are such that people don't go to out-of-house timesharing any more to solve their problem. They buy it, or connect to it.

Grad: So they use their own in-house timesharing?

Rawlings: It's in house.

Grad: They use their minicomputers, their microcomputers?

Rawlings: They are used for the in-house timesharing, which was strong, and in places still is. In-house minis, or in-house PCs. Connecting up the PCs became attractive as well, with client/server technology to help with that connection.

Grad: Even though the communication costs, by this time, had started to drop pretty dramatically by the end of the 1980s, that didn't make enough difference to keep timesharing going? Communications, you told me, was originally one of the reasons for timesharing. As communication costs drop, that makes your stuff cheaper.

Stein: But also, these systems couldn't be built in house because you didn't have enough technical people to take care of the implementation. This was the manpower problem. We had headhunters -- executive search firms -- that were making millions of dollars a month pulling people out of firms for us, because there was such a shortage.

Grad: People to do what?

Stein: To do MIS applications, to build data centers, to operate data centers, systems programming people, programmers, and all that. You couldn't find enough. That's why the data services firms did well: because they had the people. They could spread their wealth, which was their expertise, around and solve these problems, because the customers couldn't get enough people in house to do the work. Even if the computers were free you couldn't do it. You didn't have enough people. That's another main thing that happened to end timesharing: we got a lot of people entering the workforce that could do information processing, do programming, do systems design, run a computer center.

Grad: So you're basically all saying, and you seem to agree, that essentially the demise of the business was not caused because of poor business decisions on management's part, or under-financing, or those kinds of things. It was caused by the changes in the hardware prices, and things like that. Is that a fair statement?

Rawlings: The software. Part of what Stein is saying is there weren't people around to do it all. Well, there still aren't enough. What's happening is that we've got standardized, off-the-shelf [applications]. Not everybody is developing his own accounting system for his company. He now buys it. So there is this standardization.

The other problem was operating the software. Ann mentioned the business of the phone company. What would happen if we ran out of telephone operators? Well suddenly we all became operators of our own telephone system. We also became computer operators, and I hate it. I have to do backup, I have to do version control on my PC. I mean, it's awful.

Orenstein: Go to cloud-based applications.

Rawlings: Yes, okay. There might be a solution. But what happens if everybody has to become an operator as well?

Stein: But take a person of your generation today, and move him back 20 years. That person didn't have the capability to do those things, even if the machine was sitting there, because they weren't trained. They didn't have the expertise. They were doing something else.

Brook: Yes, they called the customer service people, who would take care of it for them.

Grad: Let me move that one more step and describe one of the things that has always troubled me. I'm going to blame Mike Humphries for this a great deal, and you, to some extent, Ann. This meeting originally was called a "processing services" meeting. It was going to cover service bureaus, remote processing and timesharing. But both of you kept saying, "timesharing, timesharing, timesharing." Listening to all of you, timesharing was a sort of a defining name for what you did. You thought of yourselves as timesharing companies. GEIS may be a little different, I don't know.

Brook: No, we were a timesharing company too. We evolved, but it was definitely timesharing.

Grad: That's what I wanted to come back to. Separating out the timesharing market from the other companies that were here, to what extent did you not evolve to change the nature of your business and redefine yourself? How much did you keep yourself thinking that you were a timesharing company? That's what I meant by management decision process.

Orenstein: Well I think Rick Crandall, who's not here any longer, did a great job in restructuring his company [Comshare].

Grad: It wasn't a timesharing company any more.

Orenstein: Right. And in the other companies, we all tried to do it too. And in various ways either succeeded or did not succeed in doing that.

Wyman: So you [Orenstein at National CSS] created a product that eventually was worth a lot of money with Nomad [4 GL/database system]. We created Tymnet, which became the residual value that we sold. You guys [GEIS] did electronic data interchange, EDI. So everybody did something different.

Orenstein: In hindsight, okay, one could imagine that if the software that was available – like Infotab -- or any of the other software that was available on the timesharing systems -- if it was changed around and made available on a PC, then any of these companies, National CSS, or IDC in its own way, and Tymshare, or whoever, could have become a software producer selling retail software on a PC. It could have been that way. But it didn't seem to work out that way.

Brook: We did it in the EDI area, where we kind of suddenly put ourselves out of business. Because in the old days when you just went from A through GEIS to B, which was all we were doing, we were acting as a clearinghouse. But we were also doing document translation and other kinds of stuff like that. I went from a type A document to the other guy who wanted a type C document. We'd take care of that in our system. We were doing actual content change. We weren't storing data, we were just changing the content for them so the guy down at the other end could understand it, because he had a dumb-ish device as well.

What happened when the PCs came in? We created a product called EDI PC that ran on the PC, and guess what it did? It did the translation. So this guy knew, yes, I've got a standard form, and I translated it to what I want to do, I'll send it up here. All GEIS did now was literally the Internet function, that says, okay, I've received a document from A, and he wants it to go to B, I'll send it to B. Then B received it and did the translation over on his side. So we essentially put ourselves out of business that way. That is why the B-to-B people started up, because all you needed then was a switch in the middle, where you go direct addressing if you wanted to, if you were smart enough. So that's what happened. You had the compute power at each end, plus the intelligence that allowed you to directly address wherever you wanted the thing to go to very cheaply.

A Different Business Model

Grad: Of course, the Internet has such an incredibly different characteristic. It wasn't itself a business.

Brook: Right.

Grad: And yet it killed the need for our communication businesses. At least that's what I perceive. Maybe I'm wrong.

Brook: It killed the need for private networks, which we could have survived without quite happily, I think, if we just had something free to get stuff to and fro.

Bayles: We loved private networks. We would have loved the Internet.

Grad: So the point is that we love the Internet, and yet because of it you don't need your communication skills, knowledge, capability, management skills. You don't need them anymore because this crazy model came in that made no sense to me. I couldn't believe it was ever going to work. How could you possibly do it? There's nobody managing it. It's not a company. Who's running the thing?

Brook: Oh, there are a lot of people managing it. It isn't that simple.

Grad: But seriously, the profit is not from the Internet itself. The profit is from things that people do with it. Isn't that quite different?

Wyman: The profit is from providing a solution to the end user. How that solution is implemented is irrelevant as far as the end user is concerned. All he wants are his results.

Grad: So, picture if the telephone system had evolved this way. People somehow constructed all kinds of little local telephone networks, and connected to them, and nobody had to pay anything for them.

Bayles: That's the way the telephone business came about. You saw wires strung in one section of New York that didn't talk to another section of New York.

Grad: Yes, but who paid for that wire to be strung, and how did they get their money out of it?

Humphries: They sent you a monthly bill.

Bayles: They charged monthly fees for the line.

Brook: The difference is that what pays for the Internet today is advertisements. How would you manage if you picked up your phone, and it said, "This is Miss Daisy's Lingerie store, can I have whatever it is?"

Grad: Oh, you don't have that?

Brook: Not on my phone, I don't. But on everything else, yes, there is advertising.

Grad: To me, it's just how this thing evolved. The second issue with the web is that these services that we have available just sort of evolved and happened, and they're not directly paid for. They're peanuts, you know.

Rawlings: But any decent sized organization is paying a lot to its network folks to run a virtual private network. Network people are still well employed doing network stuff, and they worry about it all the time. And they are much more concerned about security than we were, I think, because we had a private network.

Brook: If you have a direct connection, you don't have to worry about security to the same extent. Slightly, but not even close to what you need for public networks.

Rawlings: It's not that the Internet has made it free for companies to communicate. They still have to pay a whole lot to communicate.

Grad: That's an interesting question, though. I just don't know the answer. Corporations today have their own internal networks. I'm sure they are secure, and as private as they can make them. Is that correct?

Rawlings: But they all connect to the Internet, and that makes things good and bad.

Wyman: They're all behind firewalls, and DMZs, and whatever you want to call it.

Grad: But they're not paying one of your companies to provide that communications network for them.

Rawlings: They're paying somebody.

Wyman: They're paying an ISP [Internet Service Provider], a phone company, or someone else. An Internet Service Provider. You pay by the kilo-characters transferred, right, to the server? You pay for your side of the pipe, generally.

Stein: I pay my cable company to rent me the line.

Grad: Did any of you become ISPs? Did any of the timesharing companies become ISPs?

Wyman: No.

Grad: Did CompuServe come close.

Humphries: CompuServe came close, yes.

Grad: And you can speak of the AOL evolution as being in a sense that kind of thing. But they weren't really for corporate use, primarily. Weren't they for individual use?

Bayles: We used CompuServe and Salesnet -- not Salesnet -- as our communication carrier.

Brook: What's happening, I presume here as well as there, is that the phone companies had to become ISPs. There used to be little companies set up as ISPs; now the phone company, certainly AT&T and Verizon, I presume they're over here.

Grad: And the cable companies.

Brook: My phone, my good old POT [plain old telephone] service went on the blink, I don't know, five weeks ago. I called up Verizon, and told them this. I check, and it's an outside line problem. "We'll have somebody there sometime in the next four weeks." Now if I wanted FiOS [fiber-optical service], bam, the guy would have been there immediately. If I call up the personal repair service, it goes to FiOS. I said, "No, no, I've got plain old copper." "Oh, well, I have to put you through to somebody else. We'll see you in four weeks time." But if you go FiOS, then they're your ISP.

Haigh: I think what's happened is that all the pieces are still there, but with the change in technology they don't have to be bundled together to make the same kinds of business any more. I mean, in the old days if you offered an online application, then you would have to have an all-purpose timesharing system on which it could be provided as a value-added option. Then you would need to have a network, your own network, to connect people to it. Nowadays you don't even have to host it yourself. You can just develop it. You have specialized Internet hosting companies that will host it for you, and everyone can pick whatever ISP they like and connect to it through the miracles of the Internet.

There was a popular business book, I think called "Blown to Bits," that came along when the Internet was new and everyone was excited. The basic point was very good: that there are all kinds of areas where just because of the technology that was used, like newspapers, you have fundamentally disparate pieces that had to aggregate together as a business model. In

the future you'll get all the same pieces, but they won't necessarily aggregate the same way to make the same businesses. I think that's clearly what's happened here.

Grad: Well it's been certainly a dramatic change, hasn't it, that's taken place here?

Orenstein: That's an understatement.

The Legacy of Timesharing

Grad: I'm ready to move ahead to the legacy of timesharing. To my mind, the change in this area has been so incredible and so dramatic, between the Internet and the web. It's a utility, but it's sort of a semi-free utility, in a sense. Not completely free, but semi. So your timesharing stuff went to hell and of course it's never been heard from since. Well, maybe not. What are the legacies?

Orenstein: Touch typing.

Grad: Touch typing. Everybody is now a touch typist.

Wyman: The legacy, I think, of this business is "software as a service".

Grad: Okay, tell me why you think so.

Wyman: That's traditionally what, I believe, the timesharing companies were providing. They were providing a service which offered software that fulfilled the need of an end user without them having to worry about managing their own infrastructure. Salesforce.com is a classic example. They're our legacy.

Grad: Is there a direct connection? I'd like legacy [to mean] not just that they're doing something now that is similar to what you did 20 years ago. Legacy means that there was a direct connection, that I can trace a path through what you did and how it evolved, either by technology or marketing or business model. Why do you think that the current "software as a service" came out of the timesharing model?

Wyman: Well, the business model, essentially, is the same.

Grad: And you think they inherited that from you? Or did they come up with that on their own?

Wyman: I think they inherited it from us.

Grad: Ah, interesting. Does anybody else agree with him?

McDonald: I don't know, I wouldn't say I was much of an expert on this, but it seems to me that it's more of a rediscovery; that people have sort of rediscovered this model independently, perhaps.

Grad: Did any of the companies that were there 20 years ago, that were doing this kind of work, go into "software as a service" as a business?

Wyman: I maintain that maybe IDC is doing that.

Grad: Okay, a good example. So they're providing software as a service.

Wyman: It's more than software. It's typically software bundled with data.

Grad: Okay. What other legacies do you think have come out of timesharing?

Rawlings: Well I think virtualization, in a number of ways. That's one of them. I was talking with Dick [Orenstein] about it. EMC and others with storage area networks are really what the disk farms were under CP. People don't see it that way. Their separate machines aren't operating on separate disks unless you buy a storage area network, and then there's one which can be shared, and it's virtual. You think you're addressing your own file, from your own machine, but you're not. Then you go to virtualized machines, where you've got a piece of some large UNIX machine, and you think you've got your own machine. It's a virtual machine that PCs run, in a way. Multiprocessors are running virtual machines.

Grad: So you connect virtualization with timesharing.

Rawlings: Well, I'm saying that to a certain extent the technology that allowed IDC and NCSS to get going was virtual machines.

Grad: But that was an accidental thing, maybe.

Brook: That was the architecture.

Bayles: Look at companies like Citrix that provide, essentially, a virtual machine running on a server, and effectively a dumb terminal, even though it's a PC. You have word processing,

or spreadsheet applications offered by Google, on their machine. All you are at this end is a dumb terminal. It's the same with online TurboTax.

Grad: Is that cloud computing?

Bayles: Well, for Citrix and Google word processing it's a definitional issue.

Grad: Tell me what you define it as.

Bayles: If you define timesharing as a relatively dumb terminal talking to an intelligent processor, then Google's Word and Excel equivalents are the same thing.

Grad: The software is sitting somewhere, and you have access to it through a terminal.

Bayles: The software is sitting somewhere and you access it through a relatively dumb machine.

Rawlings: Part of the value proposition is that, as a user, I don't have to worry about version control, and I don't have to worry about backup. I don't have to worry about a whole bunch of stuff.

Humphries: And you can share.

Rawlings: If I want to share it. On somebody else's machine I do my Google Mail. I used to keep all my mail locally on my PC. But now I don't, I keep it all on Google, and I don't pay squat for it…and the amount of storage! I can get to it from anywhere, from any PC. But also, they back it up, and they restore it when I lose it.

Grad: You did all that as a timesharing company.

Rawlings: That's right.

Grad: Would you now make a bet, just for fun, that obviously the economics can be changed to a point where that will no longer be a sensible solution?

Bayles: Probably not.

Rawlings: They have swung it. There are groups that are saying [to] do it in house rather than out house, so you've got in-house clouds now.

Bayles: Yes, well, right.

Rawlings: Local clouds. And there are also other models, other pieces that have come and gone. Certain companies have gone into the cloud business and out of it in a similar fashion to our coming and going. Some of those were less successful than us coming into timesharing.

Grad: Application systems providers – are those all in this same mode, or not? Is that just another name for these things?

Rawlings: Yes.

Grad: ASPs is the other term.

Humphries: Yes, pretty much.

Orenstein: Yes, but the Internet has made you able to access any application, by application, where you want. If I want to work on word processing, I wind up at Google. If want to do something else, it doesn't have to be at Google. It can be at some other company, and you get there just as easily.

Do We Learn From History, or Do We Reinvent?

Grad: And we think, because of all these things you don't have to do any more, that it will be cheaper. Now I'll go back to Chris' question. Is that just a reinvention, or is that an evolution? That's a fair question.

Bayles: It is a fair question, but I'm not sure that it's not moot.

Grad: No, it is. For example in the area of software products, we've always said the people who came up with minicomputer software paid no attention to what we had done in mainframe. People who came out with microcomputer software paid no attention to what we had done in either mini or mainframe. They created new businesses. They made the same damn mistakes we had made 20 years earlier, or 10 years earlier, and they're still making them in the new businesses that are starting now using ASP. They haven't learned a damn thing from your experience, it seems to me.

Haigh: Yes. But I think you can see also continuity in terms of the user experience. The companies are so short-lived. You don't need, necessarily, for one of the timesharing companies to be a leader in online software as a service. But if you think of it in terms of the user organization, they would have had mainframe applications that they would have hosted in

house and accessed through terminals. Then they would have been caught up in all the enthusiasm for client/server. Then they would have seen the drawbacks of that in practice, with all the work you had to do configuring the desktop machines, getting the right versions of the clients, getting the ODBC [Open Database Connectivity] links, keeping everything up to date.

Then they would have had the experience that all companies have had through the computer age. People are always annoyed because the IT budget is always going up so fast, and nothing is ever ready when it's supposed to be. That would prime them to be receptive to the idea: you know what, maybe we could go back to the good old days when we didn't have to do all this supporting of the desktop configuration and the complexities of client/server. But without having to go back to having our own enormously expensive mainframes with their own support staff either. Maybe it's time to give Salesforce.com a go. So you've got that kind of collective learning and readiness on the user side.

Grad: As you say, it might be user-driven. That's interesting.

Haigh: And it's all happening in one person's career, so the learning and the experience are inside the heads of the people who are consuming the products and giving them that kind of readiness.

Grad: Ann, you haven't had a chance to speak yet on this. Do you think what he's saying makes any sense?

A. Hardy: From what I heard, it sounds like I agree.

Luanne Johnson: Yes, because we've always talked about this from the perspective of the companies themselves. We've complained so many times that these young guys just never paid any attention to all of our experience and went out and did it again. But Tom's point is that from the user's perspective there has been continuity.

Brook: To me, one of the things that is exactly down-the-line straight evolution is this: I go and do online banking. I'm taking my PC, which is actually not a dumb terminal, it's a smart terminal, because it's doing my own stuff here when I want to. But essentially I'm going to my bank to check my bank account, which is what I used to do online with Mach III or whatever, depending on who your bank was. Download information if you need to, and then update your local files. So it's very similar.

Grad: From the user standpoint it's the same experience.

Brook: It's exactly the same experience. I could go to Bank of America through using your system back in the 1980s. It was not quite so pretty, but I could download the information, and update it myself.

Grad: Now the question is, did the users looking for that experience drive the producers -- the suppliers -- to produce these products? Or did the suppliers do it on their own and then look for a market?

Brook: I think both. I think that the users expect it, and the bank loves to do it, in this particular case, because it's cheap. They can get rid of staff. It's the same idea we had with timesharing. Timesharing saves you people.

Grad: The bank has its own software to do this, doesn't it? Each bank has its own software.

Brook: Yes. Each has its own software, which it provides. But that's a one-time shot, updating the software. But you don't have the staff.

Grad: I have 30 different banks, each writing their own software, instead of all the banks using some common software.

Brook: Yes, you probably buy it, you get an OEM license. It's web front end software.

Grad: What are some of the other legacies that have come from the timesharing experience? We've mentioned these: software as a service, applications system providers, and cloud computing. Give me some more.

Brook: Well that last point can go a long way. Back then, I know our people expected to go from anywhere in the world back into their data, and be able to do work. If I go to Japan, I can do work. If I go to France, I can do work. Today with the Internet you can do exactly the same thing. So yes, I can still get to where I wanted to go to.

Grad: So I'm not location dependent any more.

Brook: Not location locked.

A. Hardy: I think that Jeff's point, again, is that the user expects to be able to do this. It's not something you send off and have somebody else do. Timesharing taught people that they're the ones that can do this.

Grad: They're in control.

A. Hardy: You're in control, yes, and it's not something you delegate.

Grad: That has conditioned the user to look for certain kinds of things, and be willing to buy them that way. So we're willing to do all their work for them on airlines reservations, and all of our banking, and everything else. We're doing the work for them so they don't have to have a customer service person.

Rawlings: Or we're unwilling to turn that over to somebody else.

Grad: Oh, I'd love to, if it was easy! Seriously.

Rawlings: You want to turn over to somebody else your bill paying? No, you don't mind having them put the stamp on the envelope, or whatever, but most of us, at least, wouldn't want them to decide which bills to pay when.

Grad: A lot of us are doing automatic bill paying. Not me, but a lot of other people are all signed up to get the bill paid automatically.

Johnson: Yes, but you decide which ones you want to have paid automatically. It’s totally under your own control.

Grad: Interesting issue. When you do that, do you look at the bills with the same care, the same thoughtfulness as when you pay with your own check?

Orenstein: You do when they're different.

Johnson: I do.

Wyman: You can say, "Don't pay this bill if it's over a certain amount".

Rawlings: And I can see it on a regular basis. I don't have to wait for a month to see what my bill is going to be.

A. Hardy: Yes.

Wyman: Or wait until your power gets turned off.

Rawlings: Or get my power turned off.

Grad: Any other examples of where you think there's an inheritance?

Wyman: Yes, with "software as a service." Let's use Salesforce.com as an example, because my clients are using them, and I encourage it. Salesforce.com offers largely free training, kind of like we did in the timesharing era. They also are all over you to see how happy you are, because they have a recurring revenue model. They have a lot of the same customer support, customer service policies, and pricing that we did, because they're making their money off of resource units. Basically, that's how they're making their money.

I see a total turnaround. We did a lot of things right at Oracle, but we were obsessed with quarterly revenue. So a lot of customers got ticked off, because we'd sell them the software and then we wouldn't be very receptive after that until they got really ticked off over a certain threshold and then we'd be back. Software as a service companies don't do that. They're just like we were in the timesharing era. They know that if somebody gets a little bit ticked off, they might lose them and they might not ever get them back again, and they need that monthly money. So the attitude has changed back to the way it was when we were in the timesharing business. We believed that it was our job to make our customers happy, and if we did, they would pay us money and everybody would be happy.

Email and E-commerce as Outgrowths of Timesharing

Grad: Two other things occurred to me from discussions here. One was that some of you created the e-mail model. Of course it's now being delivered in a different way, but that's an incredibly significant application for most of us, and for most businesses. I think that's a legacy of what I believe the timesharing companies and their outgrowths did. Do you agree?

A. Hardy: Yes, definitely.

Grad: One could argue that e-commerce is the name of the game today, even though, again, it's done differently from what we did before with EDI, which was a community thing. Now it's everybody, whether it's person-to-person, whether it's business-to-business, or person-to-business.

Belvin: It was ARPA, not timesharing service bureaus, that invented e-mail in RFC-822. [A "Request for Comments" memo in 1982]

Grad: Well you started it.

Belvin: It's ancient.

Grad: Okay, but didn't some of you provide an e-mail type service?

A. Hardy: We had e-mail, right.

Grad: So that's another step.

Belvin: ARPA was the place. The UNIX university community was a big first user of RFC-822. That's where the love of e-mail, I think, blossomed.

Grad: Well, IBM might argue that they had PROFS.

Brook: PROFS, Yes.

Grad: There may have been others, but that's the one I knew. Everybody used PROFS, and that was the way you talked with everyone else. It was another model, but it certainly contributed to the popularity of email.

How about e-commerce? My argument is that your kinds of services led to people doing the transactions, the business-to-business, to start with on line. In some sense a legacy of that today is that everything is done on a communication basis.

A. Hardy: I think a lot of companies tried things like that using timesharing. The airlines tried out online reservation stuff with us and maybe with others. The banks tried out online bill paying back on timesharing. There was a lot to learn before you can do what they do today.

Grad: Interesting thing that you talk about the timesharing model, because SABRE [the American Airlines reservation system from 1960] was not a timesharing system, was it?

A. Hardy: No, It was not timesharing.

N. Hardy: I would define timesharing as when the person at the terminal controls which program he is talking to. Certainly an airline reservation system was not like that. The airline control is one system, one program.

Grad: But the person at the terminal is deciding whether he's looking for a seat, or looking for a plane, or looking for a price, or looking for a record. I happen to agree with you: he's talking to one application.

Brook: But so are you today. You're talking to one application. If I go to United, I go to United, that's it. If I wanted to go somewhere else, I can reconnect, essentially. That's what we did in timesharing. We issued a "hello" command, and went to a new place. Just put a new ID in, and off you go.

Grad: What if I'm talking to Travelocity?

Bayles: SABRE was installed at airports and at big travel agents with dedicated terminals. You could disconnect it, and plug into United.

Brook: But there were interactive versions of it, too. We had one.

Bayles: No, later on. I'm talking about SABRE in the mid-1960s

Brook: No, we were running SABRE stuff in the 1970s and 1980s.

Bayles: But I'm talking about in the mid-1960s.

Brook: Oh yes, 3270 [an IBM interactive terminal] type traffic coming out of there.

Timesharing Influences on Networking

Grad: What other major applications, functions, and so forth, in effect were tested, tried, and proven to users as a result of the timesharing experiences?

Wyman: Networks. We didn't cover that specifically, did we? Aren't today's networks basically a product of what had to be done in order to have timesharing services back then -- Tymnet, and the other networks, GEIS' networks?

Grad: Okay, is that valid?

Brook: Absolutely.

Grad: What else?

Brook: It was different technology with the same effect.

Bayles: Was any part of the Internet implementation effort affected by X.25 stuff? Or was it a completely fresh start?

Grad: That wasn't an X.25 connection, was it?

Bayles: The question is whether X.25 influenced the Internet. I understand it certainly wasn't an X.25 protocol, but the fact is that we had X.25 networks running by 1972 or 1973…the question is, whether that influenced the Internet

A. Hardy: I think it was before that.

Bayles: And I don't know the answer.

Brook: X.25 was just a straight replacement for a phone call.

Bayles: Yes.

Brook: It was government-mandated, because the government said: We're putting up our own X.25 network. You will use it or else. And it was cheap. Free in some countries. So it was just a phone company replacement.

Grad: Interesting question, though. X.25 was a government-mandated thing, right?

Brook: In some countries. Not here. But in France it was.

Grad: Does the Internet ever get to that position, where it is government mandated? Or did it just sort of evolve from the ARPANET?

Haigh: But it's the other way round. The Internet was actually supposed to migrate over to X.25.

Grad: Is that right?

Haigh: Yes, and switch from TCP/IP. There are RFCs that describe planning for that, and testing, and efforts to get the things running. But then, in the end, the world just went the other way. Everybody migrated to TCP/IP, and the Internet linked everything else. But that was not because anybody ever ordered it to happen. If you look at the RFC documents, there are RFCs dealing with transitioning the Internet to the OSI-ISO protocols including substituting X.25 for TCP/IP. So even in the maybe late 1980s, early 1990s time period, the people running the Internet thought that they were going to switch to X.25.

A. Hardy: Yes, right.

Brook: I'm not sure the RFC said that. I mean, there were RFCs written for every possible thing you could ever dream of. Like the French, they've probably done this. "We have mandated you use X.25, therefore here's an RFC to say how you do it", rather than, "You will do this". There were a lot of RFCs written for feasibility, not for "you will do it" or "it will be done this way". Whoever did that, it never happened.

Grad: In the last two minutes, is there anything else you want to claim as a legacy of the work that you did, besides the seven things that we've already described here? Anybody else? Mike, do you have any more to add?

Wyman: I was thinking about something that Jeff said a little bit earlier, and whether it's a twist on that. The timesharing companies basically, I believe, were the first to take the step of allowing or enabling a regular person, to some degree, to be able to access the power of computing. Before timesharing, everybody that used computing, pretty much, was either a DP [data processing] person or something like that, right?

Grad: You had to go through DP, through the guy who controlled the glass house, and told the programmers what to do. That's a very good point, Mike. I like that.

Brook: Especially when you carry it on down to the consumer. Yes, the consumer can do it. Whatever you call it, the home market now, which we had with Genie, which we had with AOL, and so on. That really put it in everybody's hands. And now we have texting!

Wyman: Now you hear grandparents bragging about the three year old kid who is on the keyboard. You also hear grandparents bragging about the fact that they can now send e-mail and all. So this thing has gone from just the ordinary person in middle age, to both ends of the spectrum as well. It had to start somewhere, and I kind of think it started with timesharing.

Grad: That's a very good point. Again, it was relatively professional people using it, but they weren't professional programming people, they were professional in something else.

Wyman: I had customers that you couldn't even call relatively professional. <laughs>

Haigh: So you can argue for this as being the beginning of personal computing if you think it's a way of interacting with a machine, and having a personal relationship with it. And it is.

Stein: Yes, well said.

Grad: I thought that somebody was going to mention that. I was fortunate enough to have worked in 1954 on the Univac I while I was at GE. To do my debugging I had the entire machine to myself. It was my machine. I had an operator there because I didn't know what the hell I was doing, and it still took me three months to debug the programs. But I think it's exactly to the point that timesharing made computing a personal experience. Then when the PC came out, that was the closest you could come to what had been that timesharing experience. I think it's one of the wonderful things that you all gave to the world.

I thank you all, and that's the end of our session.