Timesharing/Professional Services Workshop: Session 1: Technology – The Early Years

Moderator:
Burton Grad

Recorded: June 2, 2009
Mountain View, California

CHM Reference number: X5386.2009

Table of Contents

Timesharing/Remote Processing Services
Session 1: Technology – The Early Years

Conducted by Software Industry SIG – Oral History Project

Abstract: After the attendees introduced themselves and gave a brief summary of their backgrounds, the discussion focused on the technological foundations for the timesharing and remote processing services industry. The participants described the initial developments which made timesharing a practical operating system approach. They reviewed the technological breakthroughs in hardware and software that made timesharing and remote processing services economically viable approaches. They covered the differences among the various developments. Primary focus was on the development at UC Berkeley with SDS which led to the formation of Tymshare and Comshare and the work at the IBM Cambridge Scientific Center on CP-40 and then CP-67 which led to the formation of National CSS and IDC.

Participants:

Name	Affiliation
Burt Grad	Moderator, SI SIG co-chair
Dick Bayles	National CSS
Frank Belvin	Interactive Data Corporation
Chris Brook	GE Information Services
Rick Crandall	Comshare
Ann Hardy	Tymshare
Norm Hardy	Tymshare
Mike Humphries	Tymshare
Paul McJones	Adobe, CHM volunteer
Gary Myers	Tymshare
Dick Orenstein	National CSS
Nick Rawlings	National CSS
Ken Ross	Ross Systems
Dave Schmidt	Tymshare

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
Ed LaHay	SI SIG member

Burt Grad: I'm Burt Grad and we welcome you to the Computer History Museum. It is June 2, 2009. I'm co-chair of the Software Industry Special Interest Group, and Luanne Johnson is the other co-chair. I just want to recognize some of the people who helped make this meeting happen. Mike Humphries has been of great help as has Ann Hardy; and Nick Rawlings has been the co-chair working with me on all of the preparations. They'll introduce themselves later on when the rest of you do. Please turn your cell phones off -- not on vibrate, but off. If you miss a message, you can catch it at the break at 10:15. Doug Jerger is the other person who has been doing a ton of work on all this, and is our yeoman helper on the Software Industry SIG. Doug will also be taking photographs of you at various times for posterity. Historians love to see photographs of people at the meetings. There'll be a couple of other people joining us as the day goes along. Some are delayed a little bit for travel reasons; some are coming in this afternoon but will be with us all day tomorrow. Chris MacDonald is a historian from Princeton University, and he and Thomas Haigh, the other historian in attendance, will be working with us during this session. Unfortunately, there's always a bunch of administrative details. We have electrical plugs under the table. So if you really, really need it, you can plug in and use your computer. It's not illegal.

The Software Industry Special Interest Group is part of the Computer History Museum. Luanne and I started its predecessor, the Software History Center in the year 2000. So it's almost 10 years that we've been doing this. This is the 10th meeting in the pioneer workshop series that we've done, starting in 2002. And Rick Crandall who is here was at that meeting on ADAPSO in 2002. Since then, we've done two meetings on PC software, two meetings on minicomputer software, three meetings on professional services and one on relational database management systems.

The purpose of these sessions is a very serious one, although we try and keep the tone friendly and light. We're trying to capture your recollections. We're trying to capture what you still remember, what took place in many cases, 30 or 40 years ago. Some of you will have different memories. "He didn't do that," "It was unimportant." "She couldn't possibly have done that, I

know better," that kind of thing. Part of what historians like though is that you get this variety of pictures, variety of scenes. It's the elephant story all over again. Each of you sees different parts, and when it is put together as a picture, they get a different insight, a different view. We would love to have all of you sit down and actually write your memoirs, but we have not had much luck in getting most of you to do that. A few have, and they're very valuable. But this is another way to get your recollections documented. Another thing this always does which is fun, is that many of you say, "Hey, I can't possibly remember what happened then," until somebody else mentions something and you say, "Oh yes, I remember that, he's dead wrong, this is the way it really was." We're preserving these recollections. I don't know whether historians in this generation will use it, or whether it'll be 20 years or 50 years or 100 years from now. But that's why we're working with the Computer History Museum, because this is a permanent place for that material to be stored, archived, and available.

As you see, you're going to be on camera. And as each of you speaks, the camera will pan to you so that when the recording is transcribed, the transcriber will know who was saying those particular words. We then take the transcripts, we edit them, and then we post them on the Computer History Museum website so they're available. Historians will want to look deeper, they'll want to have more information, look at some of the collections that are here, the actual materials. But this is often a very good starting point and gives them insight as to how an industry developed and matured, what its problems were and how they were solved or, in some cases, not solved. You've all signed releases, so that means we can use anything that you say here. We do edit out any profanity or statement that's inappropriate. We edit out anything that's libelous, because we don't want to get sued. But other than that, it's fair game.

I will moderate the sessions for the next two days. And of course I have your names here, and I will try and remember who all of you are during that period of time. There are a couple of minor administrative things, some of which I've mentioned already. Please, you are on camera, and your voices are being picked up from the mikes all around here. Do speak up so that we can all hear you. This is a good-sized room, acoustics are reasonable, but still, you have to speak up. If you turn toward me, those in the back won't hear you. So talk to each other, not to me.

Secondly, don't shuffle papers even if you want to look something up. I had a meeting recently, the one with Government Professional Services. And there was one man who had brought a ton of good stuff; he was a terrific guy with great information. But he was turning the pages much of the time, and it seriously affected the quality of the transcripts. In these sessions we're looking more for what you remember. If any of you brought things to be donated to the museum, go to see Doug Jerger and give them to him; he has forms for you to sign to take care of contributing your papers. Please do not carry on side talk. I know how important it may seem, but don't whisper since the mikes will pick it up. And when one person is talking, please let him or her finish. The range of topics here is incredible. We have organized the meeting into a series of sessions that are relatively chronological. But as we cover technical or business

subjects or marketing and sales, we're going to flop over of course. And that's not a bad problem. But in some cases, we will say, "Look, let's wait until this other session, so we can explore that in depth." And there's one other issue, and it's not a big one here. We have a number of companies represented, and a couple of companies are heavily represented, which is wonderful. They were obviously the most important companies in the field. But some companies only have one representative here, so let's make sure that we give everybody a chance from all the companies to speak. And for those where there are multiple representatives from one of the companies, each of you may have a different aspect from Tymshare or from National CSS to talk about.

John Hollar, who is the executive director of the Computer History Museum, and Len Shustek, who is the chairman of CHM, would like to welcome you. Hopefully, both of them will have a chance to drop in sometime during the sessions and greet you. John will be joining us for dinner tonight. All of you, except for anyone who said they weren't coming, will be having dinner at Ming's Chinese. In your folders, there are directions on how to get there. It's right up 101 and is hard to miss.

There's a lot to cover, so we'll move ahead quickly now. What we always start with is asking people to introduce themselves briefly. Now, thanks to Nick Rawlings, there are bios of each of you in your packets. So do not go through your full bio. We can read them. There will be a time to talk about your specific projects and things you worked on. Now we just want to have your names, principal affiliation, and go on to the next person. So let's start from there, and it will give a good chance for Albert who is running the cameras to make sure he can pick up your face as you speak.

Introductions

Dick Orenstein: Dick Orenstein, National CSS.

Mike Wyman: Mike Wyman, Interactive Data Corporation.

Nick Rawlings: Nick Rawlings, National CSS.

Dick Bayles: Dick Bayles, National CSS.

Frank Belvin: Frank Belvin, Interactive Data Corporation.

Chris McDonald: Chris McDonald, Princeton University.

Ann Hardy: Ann Hardy, Tymshare.

Dave Schmidt: Dave Schmidt, Tymshare.

Norm Hardy: Norm Hardy, Tymshare.

Rick Crandall: Rick Crandall, Comshare.

Chris Brook: Chris Brook, GE Information Services.

Luanne Johnson: Luanne Johnson, now co-chair of the Software Industry Special Interest Group.

Paul McJones: Paul McJones, I'm currently with Adobe and a volunteer at the Museum.

Jeffery Stein: Jeffery Stein, Online Business Systems and now chairman of the board of the IT History Society.

Ken Ross: Ken Ross, from Ross Systems.

Mike Humphries: Mike Humphries, Tymshare.

Doug Jerger: Doug Jerger with the Software Industry Special Interest Group

Grad: Paul McJones is an added benefit for our meeting. Paul had been working with the Museum for many years, and he has worked with the Software Preservation Group, which is interested in the bits and bytes and code and things like that. And Paul has a done a terrific job. We had a lunch with him yesterday and invited him to join us today. And he mentioned he has a background with the University of California TSS project.

McJones: That's correct, University of California at Berkeley.

Grad: So he's a legitimate participant and not just an observer. And I think that gets me to the first subject. I'm going to start with an odd thing, and we won't spend too much time on it. But our first session is on the technology, the early development, where it came from. And one of the things that I've had arguments with a number of you is the line between timesharing and remote processing services. Is there a line, is there a difference? The timesharing experts are all convinced that timesharing is the only thing that's important that happened as far as remote access and remote usage was concerned. There are one or two people who disagree with that.

And we can start around with almost anybody who'd like to start it. Who's a bigot on one side? Mike, you want to start?

Online Transaction Processing vs. Interactive Computing

Wyman: I'm not necessarily a bigot, but my recollection is that remote processing services primarily refer to batch servicing, where somebody had a batch terminal, an [IBM] 2780[1] [The IBM 2780/3780 were Remote Job Entry (RJE) stations, comprising a printer and punched card reader/punch and used mainly for interacting with a mainframe computer system] or something like that, where they fed in a card deck and had it processed and got back printed results of some sort, whereas timesharing referred to interactive use of a remote computer.

Grad: So, would you draw a line between online transaction processing and interactive use, or not?

Wyman: No, I would not do that. I would say those were the same thing, because they're both interactive.

Grad: But when I think of timesharing, I think of the technology of time slicing, of breaking up the time on the machine. Is that not what you all think about? Dick Bayles?

Bayles: I'm not sure what I would call transaction processing existed in the 1960s, or early to mid- 1970s.

Grad: How about Sabre?

Bayles: Sabre was there, that's true. It was not batch processing, it was certainly not timesharing. It was somewhere in between.

Grad: So another kind.

Bayles: Yes.

Grad: Other comments on this from any of you?

Brook: Certainly we ran both side by side. We had what we called foreground and background. And background was GCOS [General Comprehensive Operating System, formerly

GECOS] based, for those who are familiar with the acronyms, and was entirely batch processing. There was a lot of interactive ability to get into it. But the data came in from [IBM] 2780/3780 type stuff or even other host computers, and got processed separately. Then the data was able to be accessed interactively. But the data collection and distribution would be done in a batch mode. I think that was the difference that we certainly saw.

Crandall: I don't actually remember that there was such a big argument. Maybe there was later on. But when these terms first came about, I don't think there was the phrase "online transaction processing" when the timesharing technology and style of interaction came about. And my recollection of remote processing was when we started using the term at ADAPSO, and it was an umbrella term that was over both what we would call timesharing and what we would call other kinds of remote processing. So I don't think it was either/or. At least I don't remember it that way.

Grad: I guess my question is, is timesharing defined by the technology of time slicing or is it defined by the interactivity of usage? That was the question I was trying to get at.

Rawlings: I'd just say interactivity, if those are the choices.

Grad: Give me some more.

Rawlings: There was time slicing when we were getting to MFT, which was the early ability to be dividing the machine among multiple processes in a batch mode so that you didn't always run one completely to completion before you started the next. We were overlapping in the operating system's processing. But we weren't slicing it in the same way with the priority of deciding that the interactive carriage return should be worked on before, in preference to some batch system. So priority toward the online interactive person, whether he was doing a transaction into some kind of application that somebody else had written for him, or was editing and using the application development piece.

Grad: But in the broadest sense then, we're using this meeting to talk about all of those remote applications, remote usage. And yet the timesharing people really focus on the technology that made timesharing technically feasible. Ken?

Ross: I think that the key thing, because I'm an interactive bigot is that the thing that really distinguished timesharing was the ability for the end users to interact with the computer directly and effectively bypass the glass wall, or whatever it was called, in the big data center. I mean, that was always a bottleneck for end users. And then timesharing came along and end users could sort of control their own destiny. And that's really what I think made all of the

timesharing companies successful. Certainly it was so for mine, and I know it was so for Comshare and GE.

A. Hardy: I think that I'm with this interactive side, because first of all, there was a lot of technology that went into making the computer able to be interactive with the user. And it completely changed the kinds of applications that you could run, because of the fact that it was interactive. You had to have that interactive technology first. But then the things that we eventually ran on that, like airline reservations, that has to be interactive. If you just send in a request, then it's not the same as being able to look at a list of alternatives and then interact with that and ask some more questions. That's where the interactivity became so important.

Grad: So you feel that this break between interactivity and batch, regardless of whether they both could be remotely activated or started, that's the major differentiation. Rick, you had a comment you were going to make?

Crandall: Yes, I was going to say, because there's both a technology part of the definition and a user experience side. And on the technology side, I know our system, and I think most of yours, were capable of both interactive use as well as what we called remote batch. And the ability inside the system to know how to respond. And on the interactive side, there was a lot more technical work to do, because we disguised a lot of computer activity in order to be really fast at getting back to the user and so on. So there was a technology component that started with paging hardware and that sort of thing that allowed you to divide up a machine into different pieces for different users. And at the same time, we were being very sensitive to the user experience so that it would be truly interactive.

Historical Roots of Timesharing: Dartmouth and MIT

Grad: Let me use that as a jumping off point, because the thing that we tend to focus on is the development of that ability to be interactive, what we call the development of timesharing. And that took place starting in the 1960s. What we'd like to do is talk through is what were the major technological events? Who did what to whom and where? And not about the commercial use so much, but where did the technology come from, what were the important steps? What were the earliest timesharing programs, who developed them?

Brook: I think that Dartmouth's timesharing was probably the original father of it all. Which ended up being Mark 1 on the GE system.

Grad: When was that?

Brook: That was around 1960.

Crandall: Probably 1964.

Brook: Well, it was before that, because I know we were running Mark 1 back in London while I was still there, so it was at least 1964.

Bayles: Yes, it was earlier than 1964 by some amount.

Brook: I think it was about 1961, 1962, when it was developed. And then it went commercial by 1964.

A. Hardy: I think there was a lot of work done earlier than that at MIT. And I have a paper here that John McCarthy wrote describing a lot of that earlier work, which started as early as 1955 at MIT.

Grad: What was the reason for that work?

A. Hardy: They weren't getting very much use out of their computers.

N. Hardy: It was taking a day between shots.

Grad: So it was the delay in the response time that was causing the problem.

A. Hardy: The delay in the response time, right.

Grad: So he was thinking of interactive use at that point?

A. Hardy: At that time. Right.

Orenstein: Well, I think in 1962, I was working on, I think it was an [IBM] 7090, at the time at MIT. It was called a compatible timesharing system—CTSS. And we were doing word processing; "you've got mail" was a comment that was typed out on a typewriter or a Flexowriter or whatever the terminal was at the time. It was not an AOL thing. We were sending email in 1962 and 1963. There were three users on our 7090, three simultaneous users on a 7090.

N. Hardy: Yes. I used CTSS, I think it was, in 1963. You could debug and interact with machine language programs. It was a very expensive machine. The system would probably take $5 million to replicate. It was large and complex, but it did prove to a class of people that interactive access on a machine-language level was possible.

CTSS at MIT

Grad: Who led that work at MIT?

Orenstein: Fernando Corbató.

Bayles: Corbató, yes, right. IBM contributed the machine, which is the only reason MIT could afford it.

Grad: And their reason was this glass house problem or some other reason? Why were they doing this?

Bayles: Why was IBM doing it? IBM was doing it to sell machines.

.**Grad:** Why was MIT doing it?

Orenstein: To prove it could be done, that this was the way computing was going to be used, to break down that glass wall, to give people closeness to the machine.

Bayles: The term MAC started out being Machine-Aided Cognition. Ended up being Multiple-Access Computing, but the original reason for it being was a lot of the work done by Noam Chomsky and a couple of his disciples who were building, who in 1962 and 1963 had built ELIZA.

Grad: ELIZA was built by Joe Weizenbaum.

Bayles: Oh, it was LISA; LISA is the name I'm thinking of. Not ELIZA, LISA. But the original project MAC acronym, which became interpreted as Multiple-Access Computing, starting out being machine-aided cognition.

Grad: Well my point is that I think a lot of this original work may have had something to do with how people think and how people reason, and trying to provide capabilities to help them do some of these kinds of things.

Ross: In MIT in 1961 and 1962, the real interactive end user experience was being done on the [DEC] PDP-1 there. I distinctly remember playing Space War there, which they have running again here at the Museum. And there it was like a video game. It wasn't multiple users on the computer at the same time.

Grad: So it was single user, interactive use.

Ross: Totally. But it was very graphical. You could go in and change the gravitational equation and the thing would be modified.

Bayles: You had to play it with an actual fighter. They actually installed a fighter console.

Ross: Yes, right, yes.

Bayles: A joystick and everything else.

Grad: What was the purpose of the original work that was being done by John McCarthy?

N. Hardy: Well, I think, originally the problem was that what they were doing was trying to develop software on these computers. And the turnaround time for the programmers was 24 hours. And they thought if they could get a timesharing system running, which John described very early, then you could cut that time way down. And in fact, you not only cut the turnaround time down, but you allowed programmers to debug smaller increments and test in smaller increments. And it just speeded up software development enormously. And I think they realized that back in the 1950s that that might be possible.

Grad: And where was that going on?

N. Hardy: MIT.

Grad: Were these scientific applications? Were they commercial, business? Do you know?

N. Hardy: No, I don't know, but I suspect they were scientific applications. Minsky and McCarthy were both in the AI area. And Minsky had also written some papers bewailing the latency of getting access to machines.

Grad: That's interesting. So the ability for programmers to get their work done faster was a major drive?

N. Hardy: Yes.

Orenstein: This is actually a cute story that one of the other applications was chess playing. And now I've forgotten the fellow's name.

A. Hardy: Samuelson. [ed. note:Person being referred to is actually Arthur Samuel]

Orenstein: Yes. And I remember him typing for three hours and the machine crashed. And the instructions came and said that every ten minutes, we would save a file. I mean, the fact that he did not do that was his problem, but we were then told that every ten minutes, we would back up what we'd typed so that you couldn't lose so much data.

Grad: I'll stop for a second. Thomas Haigh has just joined us. He's the other historian that's working with us. And Tom has been a tremendous ally of ours over the last seven years or so that he has worked with us. Thomas, we appreciate your being here.

Belvin: Burt. If they haven't had Corby [Corbató] out here at the Computer History Museum, they're missing a big opportunity.

Grad: I believe they had. I believe that they have an oral history of him.

Belvin: That's where I would go to try to get a handle on the question that you're asking.

Dartmouth University

Grad: Now, let's look at Dartmouth timesharing. What was the purpose of the work that was being done there?

Brook: It was exactly the same motivation, but more from the students' point of view. It was all driven by BASIC, the programming language, to get them to write programs, test stuff, and turn them over fast and to increase the number of people who could run this. So you're running 60 people on at a time instead of three people on at a time. So that was their solution and GE donated the hardware.

Grad: Was GE involved in any way except later? Did they have anything to do with Kemeny or the others who did that work at Dartmouth?

Brook: Yes. Mainly, they provided the hardware and they did give some help from the R and D center. But it was mainly student-professor driven with finance.

Grad: Who else was on that besides Kemeny?

Crandall: There was another group, an outside group, called Mandate Systems. Do you remember them?

Brook: No.

Crandall: Jerry Weiner.

Brook: Oh I know him, yes.

Grad: Was that at Dartmouth or elsewhere?

Crandall: Well, he was connected with the Dartmouth project. I think he was in GE at the time.

Brook: He was in GE, yes.

Crandall: He hadn't spun off yet as Mandate. And he used to allege that he was the one that found a sequestered budget somewhere that came up with the money that allowed this project to go forward. That's what his claim was back then.

Johnson: An important contribution, finding that sequestered budget.

Crandall: Yes.

Brook: I don't know what sequestered means, but GE came up with a lot of the cash.

Crandall: Well, it means that he dug it up somewhere, but I don't mean illegally.

Grad: So we have CTSS, we have the Dartmouth timesharing system. And now the stuff that was being done out here in Berkeley, who wants to talk about that?

University of California at Berkeley Timesharing

Crandall: Tymshare and Comshare both derived from the effort at Berkeley. When I came out to join with Tymshare, we got computer time on a Berkeley computer from two to six in the morning. And they had a tape-based system, it wasn't disk yet. So our way of saving stuff was we had to punch it out on paper tape on a 35 teletype. And that was pretty grueling. And after a while, the programs got longer and longer and we were spending more time punching paper tape than we were actually programming. My first connection with that was in approximately

May/June of 1965, when a Scientific Data Systems sales rep came in. And that's the computer, the SDS 930. I was working at the University of Michigan computing center, working on the [IBM 360] model 66-M, which was the Michigan conversion of a timesharing version of the IBM model 66.

Belvin: I never heard of a model 66.

Crandall: It became the 67.

Belvin: Ah, okay.

Crandall: And he [the SDS sales rep] told me about this effort at Berkeley. I had set up a demo for all the professors at the University of Michigan who were on the technical side. And the funny bit about that was that getting a phone line to actually work all the way from Ann Arbor to California was a major nightmare for the phone companies. They were so fascinated with the concept that they set up, they told me as I was sitting at the teletype, to do a least-squares application in CAL, the California Algebraic Language, which was created at Berkeley. They had a physical person every three or four hundred miles all the way from Ann Arbor to Berkeley to make sure that these circuits hung together. You know, it's sort of like a QOS thing today with the Internet. That was the first connection and contact, and then eventually Michigan actually gave me some credit towards a Ph.D. (that I never got) to go out to California, to work with Tymshare, Berkeley, and Scientific Data Systems to create [a timesharing system.

Grad: Ann, were you involved in that original work that was done by Tymshare on the [SDS] 930, 940 system.

A. Hardy: I was involved with it at Tymshare, not at Berkeley. Rick and I started with the same machine, where they were swapping the tape if you can imagine what this meant. And the system was at Berkeley; they [SDS] came over and sold it to us by telling us it was configured for 64 users. But of course, if you put more than one user on, swapping the tape, nothing happened at all.

Grad: What year was this Ann?

A. Hardy: 1966.

Grad: Did Tymshare exist at that time?

A. Hardy: Dave, I think you demonstrated this in the fall of that year. Did you demonstrate it in 1965?

Schmidt: I'll tell you, this is a really interesting experience, because I've learned so much just since I got here.

Grad: Dave, when did Tymshare and its name, get established? What year?

Schmidt: 1964.

Grad: But there was no timesharing yet.

Schmidt: That's right. Well, we were using the Dartmouth system. That's what energized me to get into the business.

N. Hardy: Because Tom O'Rourke was from GE.

A. Hardy: Yes. Tom and Dave were both from GE and were using the Dartmouth system. But when they spun out and looked like they might be successful, GE ended up saying, "No, you can't have our computer." And that's how they ended up on the Berkeley system, because GE had a business going and they didn't want their competition to use their computer system. And that's how Tom and Dave, and Rick too, ended up on SDS.

Grad: Did you originally use the Dartmouth timesharing environment?

Crandall: No, we were aware of it but didn't use it. The really exciting thing for us was the possibility that we could actually get our hands on one of these computers, which SDS eventually offered us in exchange for us going out and working with Tymshare.

Grad: Let me just break off one second. As far as the work at MIT was concerned, was there at that early stage in the 1960s any specific technical connection or business connection with any company? For example was IBM working with MIT?

IBM Scientific Center at Cambridge

Orenstein: Specifically not.

Grad: Ah, that's interesting. Because you pick up the IBM timesharing program and IDC picks this up later. We'll get to that when we discuss the formation of the companies. I'm trying to still go through the technologies.

Belvin: I just wanted to mention that I used CTSS when I was working at Lincoln Laboratory because MIT made it available. It was a model 35 teletype connection. And actually the first users at Lincoln were the psychology group. Doug Yntema and Oliver Selfridge were ones who were really interested in using it. I don't know whether that was because of Weizenbaum or something else. But then the rest of us, some of us engineers, ended up getting interested. And in fact, at one point, IBM made the 360 simulator available. And I learned assembly language 360 on CTSS.

Grad: So there was that connection going on?

Belvin: I was at Lincoln, the computer was in Cambridge, and I wrote programs and assembled them and ran them, simulated them.

Grad: So I'm trying to see where these roots are. So far, we've identified the CTSS root. Does that lead itself into the CP-67?

Bayles: In a staffing sense, yes.

Grad: From a technology sense?

Bayles: In a technological sense really, no. There was never any commercial intention to use CTSS itself as a timesharing system.

Grad: Okay, so we have the CTSS as sort of a beginning point. Was there ever a program that resulted directly from the McCarthy work? Did that lead into CTSS?

N. Hardy: I believe ITS [International Timesharing] came as a result of the success of CTSS. It was done very quickly, shortly after CTSS showed signs of life. And it was done on much less expensive hardware by a bunch of hackers that were proud of the fact that they could turn out something fast. And they did.

Grad: Where were they, Norm? Were they at MIT?

N. Hardy: Yes.

Belvin: I think the LISP machine came out of that. And the company, Symbolics, grew out of that effort. And I think John McCarthy might have been involved.

Grad: Is LISP considered to be a timesharing capability? I always think of it as a language.

Belvin: No, I was talking about ITS.

Rawlings: I wanted to say that there was an LGP-30 user group that this guy who was the number one guy at CSS [the predecessor to National CSS], Bob Bernard, who was a founding member of, and maybe a chairman of for a while. And one of the other people who was important in the LGP-30 group was Tom Kurtz, who was at Dartmouth. Around 1964 or so, Tom Kurtz said to Bob Bernard, "Come and look at this timesharing system." And so Bernard did and I went with him. And he said, "We ought to make this same idea of timesharing available to the engineers at Perkin-Elmer and let's do it on SDS equipment. So we bought an SDS 930 before they were actually manufactured. We signed up for them and we developed a timesharing system on the Dartmouth model -- the 930 is the Datanet-30, and the 9300 is the [GE] 235 -- on that model. We built it, ran it and it was very successful. Dick Bayles helped with that. And then Perkin-Elmer ran out of steam on that machine and got the [IBM System System/360 Model] 67. But the idea for this effort and the importance of timesharing to this engineering firm was based on the Dartmouth experience.

Grad: You've now mentioned ITS. Are there other names of timesharing systems of that period we should be putting down here during the 1950s to mid-1960s?

Crandall: Well, not quite that early, but there was a project at the University of Michigan, which felt very competitive with MIT and with Carnegie, all three of them on IBM equipment. And the Michigan project, where I was a student at the time, was in the computer center. The names that come to mind are Bruce Arden, Bernie Galler, and Bartels, who was head of the computer center. Those are the three key names associated with that project.

Grad: Did that go anywhere?

Crandall: They definitely had a system they considered their timesharing system, on what was called the Model 66. And IBM dubbed it the 66-M.

Belvin: Are you talking about UMMPS and MTS?

Crandall: Yes.

Belvin: UMMPS was the analog of CP in VM-370. And MTS was the analog of CMS. And UMMPS was developed by Mike Alexander and Don Boettner at Michigan. They were the software developers of that and they actually started with a little thing that Joel Winett and I had done at Lincoln Laboratory called LLMPS, a multiprogramming system, which we can talk about later on. But that saw use in many universities. It was a very popular system.

Timesharing Becomes Commercial

Grad: Let me go to the next stage. These are many of the technological roots. And now these technologies start to get picked up, and then enhancements and changes are made to make them useable in a commercial marketplace or for semi-commercial use. GE picks up the Dartmouth timesharing system, Chris?

Brook: Well, GE was trying to get into computers at this point. They pretty much started in the mid-1960s. And Dartmouth was a great way to get into it. GE threw some money at it and got some pretty smart people to develop a system they could then take back and make money from.

Grad: Well, GE started its computer business around 1960, however. So when does GE Information Services get started?

Brook: GE Information Net started with the Mark One as its first system. That was the commercial version of the Dartmouth system. I think it was probably about 1964.

A. Hardy: Dave Schmidt was there for all of that.

Schmidt: You're right. We demonstrated the Dartmouth system at the Cow Palace in San Francisco in 1964. We built the booth in my garage. And Tom O'Rourke was there selling. And that was the first exposure I had to commercial timesharing. There weren't very many of us doing it at the time.

Grad: Dave, at that point had you picked up the Dartmouth timesharing system?

Schmidt: Yes.

Grad: And you had come out of GE?

Schmidt: Yes.

Grad: You were trying to use it on a commercial basis to do what? Were you were going to provide a service bureau? What were you going to do?

Schmidt: I remember the sign we had up on the front of the booth that said, "A computer at your fingertips." That was the whole story. It was interactive all the way for all the reasons that have been brought up here already.

Grad: Dave, my question is this. Did you do things to the product technologically? Did you try to change it or did you simply use what they had done?

Schmidt: Well, we planned to make changes to it of course, changes to the GE computer and to the GE Dartmouth software. But that fizzled and fell apart. Then we went to the SDS system. We were very active with SDS, helping them define the system that they would finally produce for timesharing.

Grad: Chris, were you involved in any way in development at that point of the Dartmouth system?

Brook: No.

Grad: Was anybody else involved with the Dartmouth timesharing system in terms of its enhancement or development during that period? No. Okay. Rick, you came out here and were working with the people at UC Berkeley, right? Was that on the SDS 930-940? What kind of work did you do? What were you doing with the system?

Crandall: I came out in February of 1966. And I remember that Tom O'Rourke and maybe Dave couldn't believe that somebody as young as me had been sent out to be the representative of this other timesharing company. My role at the time, when I went back to Ann Arbor, was that I was going to have to be the sole person responsible for our version of the software, which scared the dickens out of me. We were working on making a disk operating system so that we could realize the promise of multiple simultaneous users and not get the little “TWs” on the Teletype paper, which is what the Berkeley system printed out every time another user wanted to take over the machine. TW stands for “Tape Wait.” So we worked on that. We also worked on a file backup system. A couple of us tackled some languages. I know I developed an ALGOL for the machine that we never really used commercially. But it was an interactive version, for interactive programming. And so we each took pieces of the project and I took what Berkeley had done and got it to where we would feel comfortable making it commercial.

Grad: Were you sharing what you did with the Tymshare people?

Crandall: Oh yes. This was absolutely collaborative. We worked hand in glove together.

Schmidt: Yes, we were two rooms apart. I mean, we were there in the same facility for a long time.

Crandall: I was working at the Tymshare offices.

A. Hardy: SDS delivered the 940 to Tymshare, and that's where we were both working on it. They delivered a disk but no swapping drum. It was still a very slow system.

Grad: And this was a 940, not a 930?

A. Hardy: That's right.

Grad: They'd already switched over?

A. Hardy: It was a 940 that was delivered.

Grad: They gave it to you for free, or did you have to pay for that?

Schmidt: Oh, we paid for it.

A. Hardy: Tymshare paid for it.

Grad: Why were you willing to have this kid from Michigan even work on your machine?

Schmidt: I don't remember how he forced his way in there.

Crandall: Well, I can tell you how it happened. SDS came to both of us, to Tymshare and Comshare, and said, "We will give you a 930 modified à la Berkeley." The Berkeley electrical engineering department was the one that added the paging hardware to the 930 that was really a part of making it a 940. And then SDS picked it up and they offered a number of months of free rent, which was worth approximately $35,000 a month.

A. Hardy: Right.

Crandall: Imagine, renting a computer. And they did the same thing with Comshare. They said that they would ship our machine in August of 1966, which is when I went back to Ann Arbor. And we got something like six or eight months worth of free rent in exchange for my having gone out to California. And Tymshare also got some kind of a rent benefit from my going out there.

A. Hardy: Yes, we probably got free rent for a little longer time.

Schmidt: I can't remember how long.

Grad: Because you allowed him to work with you, is that what it was?

A. Hardy: By allowing Comshare on the machine.

Grad: Was it worth it?

Schmidt: Believe me, we were very happy to have any help we could get and Rick was a major player.

Grad: Weren't you concerned about the competition?

Schmidt: No. Not really.

Crandall: First, understand in those days there was no networking.

A. Hardy: It had never occurred to anybody that there would be a competition between a company in Michigan and a company in California.

Crandall: Exactly. So the deal was that at the end of that effort which was in August or September of 1966 that SDS, Tymshare and Comshare would each walk away with a tape, which was the product of all we had done. And then we all went off in our three different directions.

IBM's Scientific Centers and Timesharing

Grad: Okay. Let me follow some of the other stories along here. We have the work at some point at the IBM Cambridge Scientific Center. Was it established in the mid 1960s?

Bayles: I think it started in 1963, late 1963.

Grad: Herman Goldstine was the one who was involved. I was working for Herman and he set up the Scientific Centers around the country; I think that 6 of them were set up.

Bayles: They were all tied in with what were considered by IBM to be key educational institutions, Stanford had one, MIT had one, Grenoble had one.

Grad: There was also one in Texas and one in New York City associated with Columbia University and I think one in Philadelphia. Was there a connection between the MIT work and the Cambridge Scientific Center work, other than some of the people?

Bayles: Other than people, no. If you look at the staff of the Scientific Center starting in 1965, which is when I joined them, there were really only two Project MAC people, myself and Bob Creasy. He wasn't the manager of the Center, that was Norm Rasmussen, but he was kind of the number two guy. He came out of Project MAC as did I. The rest were really, to the best of my recollection, largely IBM System Engineers from around the country. And field engineers as well. So there was no technological connection except being users of CTSS, I did some programming on the supervisor side, the principal experience being that as users of CTSS it gave us a framework for what the user experience should be with whatever was coming.

Grad: So that deals with the interface, rather that with the timesharing capability, is that correct?

Bayles: It's the human testing interface.

Grad: When the Cambridge Scientific Center was established, what was its goal?

Bayles: CP-40 was the project.

Grad: What was the purpose of that project?

Bayles: Well, there are mixed opinions on that. Norm Rasmussen has passed away, so I can't really ask him. So has Bob Creasy. So that tale may never be told. But the fundamental basis for the existence of the Scientific Centers was to put free machines in key universities and to sponsor graduate and post-graduate activity and anything else, in an attempt to generate a group of people who came out of those universities with an IBM orientation. So the Cambridge Scientific Center, when it started, didn't have any really specific purpose. Now this was a couple of years before I joined them, but I've talked about it with other people. They ended up sponsoring a collection of graduate students, including Nicholas Negroponte, who became the [MIT] Media Lab guy, who worked in the office next to me at one point, and three or four other

efforts. It's a little unclear what spawned the idea of what was called CP-40, which was the control program, the virtual machine concept.

Grad: I was working with IBM and interfacing with all these centers. And we were saying to them, "What are your projects and what is your purpose?" And my memory is that they were going to use their model 40 to be able to mimic or test other 360 models on this one machine for performance analysis. That's the story I was told.

Bayles: The ostensible purpose was that we were going to build a machine, a virtual machine. There was another IBM virtual machine project being done out of Yorktown Heights, the IBM Research Labs, called the 7044X, which was a modified 7040. The virtual machine it created wasn't a real 7040, it was a meta-language, if you will, for the 7040. So the concept was that CP-40 would provide an interface to its users that would exactly mimic a System/360, at the machine-language level, the purpose of which was presumably to monitor and measure operating systems performance. In my personal opinion, that was a smokescreen.

Grad: What was the real purpose in your opinion?

Bayles: The real purpose was to build a timesharing system.

Grad: Well, that was my question. I've never heard that explanation before.

Bayles: Well, this is, I mean I came there a year and a half after they started but I came there before we had a Model 40. We ended re-microcoding a model 40 and spending a lot of time in Poughkeepsie working on that. But Bob Creasy and whoever else came out of a timesharing environment and CTSS in particular were aware of MIT's decision to go with GE Multics, as well as IBM's failure to provide a reasonable alternative to that. So, in my personal opinion, they undertook to get funding from IBM for this project, which was ostensibly to measure operating systems performance, when in fact, it was essentially a disguise for generating a multiple-user, System/360-based machine.

Grad: That's fascinating. The IBM supported timesharing system project [TSS] started after IBM lost two projects: to MIT and what was the second?

Bayles: They might have lost Bell Labs.

Grad: Bell Labs. They lost two projects, and Dr. Lou Robinson, who was in charge of the technical support for the sale of this kind of advanced scientific stuff, was replaced at that point. , IBM management said, "Hey, we're not going to lose another one" and they appointed Watts Humphrey, and Watts never lost another sale of a timesharing system. He won 32 in a

row with a very simple solution. Whatever the customer wanted, Watts would answer: "Sure, we'll provide it."

Bayles: Right. Except there was no operating system to go with it.

Grad: That was the start of the TSS project. Did any of you ever get involved in the TSS project?

Bayles: I knew Andy Kinslow very well who was on the project. Frank, you were more involved than I was.

Belvin: Well, only as a potential customer. But we sent people up there.

Grad: There were 100 or so people working up in Mohansic, New York, and they had outside people doing a lot of the work. I think eventually that only 10 customers ended up using TSS.

Wyman: I'm amazed there were that many.

Bayles: I'm astounded.

Belvin: Have you read Melinda Varian's paper?

Grad: Yes, I believe that's what she said, because that's where I got it from. That is why I raised the question about the whether there really was a timesharing objective for the Cambridge Scientific Center project.

Belvin: Norm Rasmussen was a very clever man and you could not always understand what he was aiming at.

Bayles: He was an excellent politician.

Belvin: With a very far-reaching vision.

Grad: So he may have really had that as a goal.

Belvin: He may have. I don't know, I didn't work for him.

Bayles: Well, I did work for him.

Grad: Let's start again. CP-40 was the initial project, but when does it switch over to become CP-67?

Bayles: For CP-40 we essentially re-microcoded a Model 40 and added some hardware to provide dynamic address translation to handle a maximum of 15 users. And we got that running and probably had the machine installed in late 1965 in Cambridge. In the summer of 1965, we were still re-microcoding it and debugging the microcode. And the only access we had to a real System/360, Model 40 was at a sales office in Boston, since we didn't have our own machine. So we were trucking down to Boston all the time and couldn't test anything like relocation or multiple users or anything else. Naturally the CMS people, which was the user side, had an easier time because--

Grad: Those were the two separate pieces of the project?

Bayles: They were two totally separate pieces, although we knew about the other. But our goal was to provide a clean 360 interface, their goal was to run on a clean 360. So they had a much easier time testing it than we did. Our machine was installed in probably the third or fourth quarter of 1965. And we got it up and running fairly quickly.

Developing CP-67

Grad: I'm going to cut you off on the CP-40 and ask you to move ahead to the CP-67 project.

Bayles: Okay. The move to the 67 I think was probably triggered because Jack Arnow and Norm Rasmussen knew each other and talked often. In early 1966 IBM had just announced the Model 67.

Belvin: But the agreement between Lincoln and IBM predated the 67 announcement. It was RPQ'd on a model 66, which IBM said would become a standard product.

Grad: Was the Blaauw box in existence at that point, or not?

Bayles: Is that the DAT [Dynamic Address Translation] box?

Belvin: That was it. That was what made the 67s.

Grad: But that did that exist at the time that Lincoln Labs signed up? I mean, there had to be a point at which the DAT box existed as a physical piece of machinery, right?

Belvin: Well, it did when Lincoln got its first Model 67.

Bayles: Which was when?

Belvin: I thought it was 1966.

Bayles: TSS-360 had been announced by IBM as had the 67. IBM had people at Lincoln Labs working on either an early release of TSS or something. But there was sufficient unhappiness on the part of Lincoln, I guess, that Jack Arnow talked to Norm Rasmussen, who was the head of the Scientific Center, and said, "I've heard about your work on the Model 40," which wasn't publicized outside of IBM. And, "What do you think about doing a Model 67 version of that?" And so what happened was probably, by the end of 1966, Norm had committed to try to get funding to develop CP-67.

Grad: And that was for both pieces, the timesharing piece and the accompanying CMS?

Bayles: CMS never changed. CMS didn't care about the platform.

Grad: It was the same in either case?

Bayles: CMS didn't care. It was running on a standalone 360, no extended instruction set or anything else. All of a sudden, the cover was blown essentially, on the premise that CP was really an operating system to monitor other operating systems. Once Norman and Arnow agreed, and Norm decided he was going to push for a Model 67 version of CP, any rationale for it being used other than as a timesharing system disappeared.

Grad: And yet it was not in the product plan, TSS was the product.

Bayles: Well, IBM has a number of divisions. At that point, it was DPD and SDD. DPD was the sales division, which is what the Scientific Centers reported to. SDD was the Systems Development Division, which was responsible for TSS/360 and OS/360 and all the rest of systems software for the larger 360's. So the trick to get funding was to get the project started without raising the hackles of SDD, of the TSS/360 people. And quite frankly, I think they began to learn about it once we decided to order a Model 67. We had to do some emulation work on CP-40 to make it look like a Model 67. Once word got out that we were looking for a Model 67,

SDD got involved. And I think frankly Jack Arnow saved our bacon. Because SDD would have killed the effort, absent any pressure from customers saying, "I want this."

Grad: That's an interesting story. Let me now swing over. So tCP-67 becomes the basis where IDC picks up that work and National CSS picks up that work.

Bayles: Yes, the product was CP-67, which came out in May 1, 1968. It went into what was called the IBM Type III Library. Now the Type III Library at IBM was software that was not developed by SDD or DPD headquarters. They were typically applications that you could not rely on IBM to support by systems engineers or by anyone else. But it was available to customers if they signed a release.

Grad: Type III was treated as though it had been built by systems engineers in the field, as against being done by a central development group at White Plains.

Bayles: Which is effectively what it was. So it went into the Type-III Library on May 1 of 1968. And Frank and I walked away with the same tape; I went to what was then called CSS and Frank went to IDC. That was the starting point for both of us.

Grad: What were the primary software problems in building a timesharing system? Which of you want to comment to that?

Crandall: You mean technical?

Grad: Technical, not just money.

Crandall: Well, timesharing was a system that was running off of interrupts that were happening in real-time. Debugging was a whole new experience in comparison to prior systems because you had to sort of mentally visualize what was going on in parallel and figure out where a bug was. And there were no debugging tools initially. You were at the console, literally flipping through sequential instructions and trying to work out what hit you from what angle. It was a whole new experience in debugging.

Grad: How would you preserve everything? How would you save where you were to make sure you could start back again?

Crandall: You had to determine when something happened. Because obviously, you were looking at it in human time, which had nothing to do with what was going on in the computer.

Grad: Dave and Ann, what are some of your recollections?

A. Hardy: Berkeley had a pretty good dynamic debugging tool to deliver with their system, which came with our system; it was very helpful with that problem. I think the other problem was that communications was just in its infancy. When somebody types something on a Model 33 Teletype, how do they see that on the piece of paper in front of them instantly? And a lot of work went into that. And initially, the question was do you interrupt on every single character you get? Or do you interrupt only on carriage returns? And who provides the echo? And there was a just of those kinds of things.

Grad: Norm?

N. Hardy: Just last night, I found a website describing the operating system of the Ferranti Atlas, which was to the best of my knowledge the first highly multi-programmed system. And it describes in beautiful English, without all the buzzwords because the buzzwords had not yet been invented, how you do these things. And it's a wonderful exercise in reading what it was like for the pioneers to develop this stuff.

Grad: Well, none of you have mentioned any of this stuff outside the United States going on in timesharing. Maybe you would take a few minutes here. Was there significant work? Were you influenced by any of the work going on in Europe, in England, anywhere else in the world? Were any of you aware of or knowledgeable about any of this stuff?

Bayles: I would say aware of, yes, knowledgeable, no. And certainly in my case it wasn't relevant. I mean, the Ferranti system was considerably older; the base hardware was considerably older and considerably smaller.

Grad: Were you in England at the time, Chris?

Chris Brook: Yes, I was working for Bull at that time. In England it was De La Rue Bull, then later it became Bull GE. But Bull was the technological force. But they weren't working in timesharing. Their timesharing stuff all came about when GE came in. So they provided a lot of hardware, a lot of machines, a lot of very advanced machines. The Gamma 10 was one of the early punched card computers, but it was very smart.

Grad: How about ICL? Anybody there?

Brook: ICL wasn't doing timesharing. They were all doing data processing.

Grad: Thomas, are you aware of any significant timesharing work that was going on in Europe during the 1960s?

Thomas Haigh: Well, the Atlas project came out of Manchester University. At this point in the early mid-1960s, ICL hadn't been formed yet. I think it came in at the end of the decade.

Grad: So it was still early. Okay. What were the other software issues in building an effective timesharing system at that time? Frank, are you aware of any of these things that you felt were the real software barriers?

Belvin: Well, performance issues I think we pretty well touched on. I was going to say that user interface was to me the big challenge. We had absolutely no idea what an ordinary user wanted to see at an interface. And the legacy of CTSS with its commands, which became the basis of CMS, left an arcane feeling in our users. When we first introduced it at the laboratory, I think our biggest problem was training users and making them able to understand this strange command language.

Bayles: What were they used to?

Belvin: Nothing.

Bayles: Punched cards.

Belvin: Of course.

N. Hardy: Well, one other thing I would add to this is that there was a timesharing system at Livermore, which is where I came from. So some of the background for what works for users and what doesn't when you're interfacing to a timesharing system, we had explored already.

Grad: Well, one other question. They talk about the CMS piece, which is the user interface. Was there some similar work that went on at GE, Comshare or Tymshare on the user interface?

Brook: If they had stuff, Dick would probably know better than me, because that was kind of before my time. Talking to people, the R and D center was doing some work in that area on the whole interpersonal communication thing.

Grad: Norm, did you have something on that?

N. Hardy: I got a chance to go back and use CTSS for a week, and got some useful work done. In other words, I learned enough about the system to accomplish some useful work. And Livermore had already had an informal commitment to do a timesharing system for the CDC Model 6600. When I came back from CTSS, I brought back many ideas from CTSS which influenced just this very sort of thing: how does the user perceive the system?

Crandall: A big piece of the work was that once you got the religion of interactivity, it was then a question of going through every language and making it interactive. Interactive editor, interactive Fortran, interactive COBOL, interactive everything. Some of these things were not originally conceived with interactivity in mind. So you were kind of reinventing what these languages were in order to make them interactive. At that point it wasn't so much pure invention as it was taking a new principle and then rippling it through everything that was going to become a user application or a development language.

A. Hardy: One thing about the SDS system that was helpful in this regard was that it was divided into two parts much like CP and CMS; it had a CP part and it had a user interface. Since the user interface wasn't dealing with hardware interrupts and other hardware stuff, it was a lot easier to make changes and adjust the user interface than if it had been packaged as one system.

Brook: I guess there was a philosophical division with some of timesharing's early developers about whether you had local echo or end-to-end echo. And we opted for local early on so you didn't have to worry about every character coming back at you and all that. And that made a huge difference in terms of speed and processing power, if you've got to do local echo. That means our end-to-end echo was a lot harder.

Crandall: Half duplex and full duplex.

Brook: And I think that was an early fight that went on.

Grad: From the hardware standpoint, was the DAT box or its equivalent the key hardware that was needed to support a timesharing system?

Rawlings: I think so. I think that one of our major problems when we were running multiple users on a Univac 9300 was that memory protection didn't exist. So if some program said "A sub i equals zero" and they forgot to initialize i, they destroyed somebody else's memory including the operating system. Planting a zero in the midst of code was sort of problematic. But until we had memory protection, we couldn't deal with that.

Grad: So what was the key to the solution?

Rawlings: The DAT box was the key to allowing multiple things to be running without having to swap them all out and swap them all back in again.

Grad: Did you need virtual addressing schemes? What kind of things did you need to make that stuff work?

Rawlings: Therefore, the DAT box.

Grad: That was the answer to it.

Rawlings: Yes. Because I guess memory protection was also provided by the DAT box because you couldn't address outside because there wasn't an address for you.

Grad: Once you had a virtual address scheme, then you provided the other because it's your space, no one else can technically get into it. Is that right or wrong?

Bayles: Theoretically.

Haigh: But some kind of things you would also want for multi-programming though, right? So is there anything that you would need for timesharing that you wouldn't already want to have for a robust multi-programming environment?

Bayles: From a memory point of view?

Haigh: From an architecture viewpoint.

N. Hardy: Communications.

Brook: If you've got one, then you can step into the multi-programming era. You need the base stuff first, and then a whole bunch of things become wonderfully easy to do.

Bayles: There are other pieces of the architecture and the communications is a big thing. But from a fundamental, what I'll call CPU-memory architecture point of view, probably not.

Grad: That's an interesting connection.

Bayles: From an interactivity point of view, you need a bunch of other hardware and stuff on the outside to make sure that all happens. But from a fundamental memory protection, relocation or DAT point of view, I think that's probably not true.

Grad: My final question in this section is: Was there ever any consideration by any of these people about the economics?

Crandall: Well sure. It was all about number of simultaneous users.

Grad: I'm wondering whether the other people really gave a damn about the economics. That was my question.

Brook: I think it was market share first or getting a step into the business area. Then you're throwing money at it day one, but day two, you're very much worrying about the profits.

Orenstein: When we started, I used to say to people I didn't know whether it was a good business or not. But if it was a good business, we had to have the best product.

Grad: I'm asking a different question. Those are company views. I'm talking about the people who actually did the technological development, at the Scientific Center, the people at MIT?

N. Hardy: I think that with John McCarthy, you saw very early on that his very first motivation was better economics. How do you get more out of the machines, because they cost MIT a lot of money. And how do we get more users, more information passing through this machine faster?

Grad: I was using economics more in the business sense of what you could sell. And you're using it more in terms of how do you get more value out of the investment you're making.

Bayles: It seems to me that John McCarthy was more concerned about how to get more out of the people than out of the machine. And he was trying to improve productivity on the part of people. And if that required a machine that could do it, yes. But his motivation was getting productivity out of people.

Grad: When I use the word economics, I think business. You think in terms of using the machine, in terms of people productivity. It's just wonderful how the one word stimulates different directions. Norm?

N. Hardy: Salaries of programmers is a business issue.

Grad: Not if you're at a university. You don't pay anybody there.

Bayles: Not when you have $100,000 a month machine you have to fill.

N. Hardy: Right.

Grad: Now, one of the interesting comments that was made in a paper I've read recently was about the tradeoff ratio between the cost of people and cost of machines. At that point in time, the machines, if you didn't get them for free, cost so much money and the people were cheap. Within 25 years this had completely flipped.

N. Hardy: At Livermore, when we were arguing for timesharing, one of the arguments that seemed to capture a fragment of the opposition was that machines were too expensive to let them sit six hours at a time and compute without any human looking over their shoulders. Timesharing was a way for the physicist to watch over his computation as it proceeded so as not to waste three hours of valuable machine time.

Gary Meyers: I'm Gary Meyers and I had some early involvement in Tymshare. Not on the technical side, but on the marketing side.

Grad: You'll be contributing a lot more later on. But I appreciate your coming here.

Haigh: Burt, I just have one question that goes with this. So when historians write about this period, one thing we like to pick up on, because it seems so wacky now, is Grosch's Law, which I believe was that the power of the computer was supposed to increase as the square of the price. So the largest computer would always give the best price-performance ratio. Now, was that something that people actually believed? Did you ever hear of this so-called law at the time?

Grad: Now Herb Grosch was, I remember at Evendale at GE, and in his view the bigger the machine, the far more efficient it would be from a price-performance standpoint. That was the view.

Crandall: I tell you, we felt the bigger the machine, the more it cost you when it went down.

Grad: This is the end of Session #1; we'll reconvene in 15 minutes.