Oral History Interview with

Stuart Card, Ph.D.

February 17, 2020
Los Altos Hills, California

Conducted by Jeffrey R. Yost, Ph.D.
Charles Babbage Institute

Abstract: This interview is part of a series on Human Computer Interaction (HCI) conducted by the Charles Babbage Institute for ACM SIGCHI (Association for Computing Machinery Special Interest Group for Computer Human Interaction). HCI Pioneer Stuart Card discusses early education, attending Oberlin College, and helping lead its computer center, before the bulk of the interview focuses on his graduate education at Carnegie Mellon University working under Allen Newell, and his long and influential tenure at Xerox PARC. This includes his long and impactful collaboration with Newell and fellow Newell doctoral student Tom Moran. Newell, Card, and Moran were fundamentally important to theorizing early Human Computer Interaction, and the three co-wrote the widely used and deeply insightful textbook, *The Psychology of Human Computer Interaction.* Card provides an overview of his decades of work of Xerox PARC and various aspects of his research contributions to HCI models, information visualization, and information access (especially foraging theory). He moved into managing research and also relates a portion of his leadership roles at PARC and outside on important committees such as for the National Academy of Science. He briefly expresses his ideas on the early institutional history of SIGCHI and its evolution. Regarding his work at PARC, Card discusses his influential work on computer mice research at greater length. Card became an adjunct professor at Stanford University. He is an ACM Fellow and was awarded SIGCHI's Lifetime Research Achievement Award.

Yost: My name is Jeffrey Yost, and I am the director of the Charles Babbage Institute (CBI) at the University of Minnesota. I'm here this morning on February 17, 2020 with Dr. Stuart Card in Los Altos Hills, California. Stuart, I see you were born in 1946, can you tell me where you were born and where you grew up?

Card: Actually, I was born in 1943. I saw somebody had that erroneous information on the internet.

Yost: Ok.

Card: But it's 1943, the one year they made pennies silver. I was born in Detroit, Michigan, and we lived in Grosse Isle, I think, for a couple of years. And then, I really grew up in Romeo, Michigan. It's a small village of 4,000 people, sort of like a classic small town of a generation before. Except for the fashion of the cars, it looked like it did in 1910. It was exactly like the town described in Thornton Wilder's "Our Town". My mother went to high school there. So by the time I was graduated from high school we had been there for 30 years, but we were still considered newcomers.

Yost: Were there—in junior high, high school—subjects you especially liked or special aptitudes you had in particular areas?

Card: Well, there were two things I especially liked, I always liked science and debate. So, my extracurricular time was focused on debate because I was a debater—I made it to the state quarter finals—and it was also focused on the science fair. In Ninth Grade, I built a telescope, grinding my own mirror—

Yost: Oh, wow.

Card: —that sort of thing. Then I tried to do an electronic diffraction device that I saw in *Scientific American*. But I could never quite get the vacuum low enough. What you do is you take some old television tubes and cut their necks in half—and there's where the electron gun is—and you try to put gold foil or some other target into the tube in front of the electron gun. Then put it back together and, pump all the air out, fire it up, and you should see the diffraction patterns on the tube face. Now, it has to have a pretty low vacuum, so lots of luck getting the vacuum that low. I finally discovered I needed oil diffusion pumps, which was not a fact or a resource easy to discover from a small town, but the real fatal blow was that I needed special oil diffusion pumps over multiple contiguous days of pumping, and that was far beyond my resources. From First Grade on, I had a best friend, Jim Phillips, who in high school also became my debate partner and science fair colleague with his own project. He always seemed to know stuff I didn't, like existentialism and Dada art and marionette theater and chemical oxidizers, and he would teach me about them. He's a professor at Columbia now.

Yost: I went to Macalester College, which is kind of like Oberlin, and I know both of those schools have very successful debating programs. Did that have an influence in your choosing Oberlin College?

Card: No. Actually, I wanted a liberal education but one in which I could get a real science degree.

Yost: Ok.

Card: I was interested in going into Astronomy, so I was a physics major.

Yost: In your last year at Oberlin, you worked as a systems programmer and assistant director of the computer center, and the following year, after graduating, you became acting director of that center. Can you describe that center to me and your work?

Card: It starts out—Oberlin didn't have a computer when I went there. So, I don't know how it happened for the college, but they finally did get a computer, a little one, an IBM 1620, which didn't have a disk in it. Everything was done by punched cards. In order to compile a program, you would have to hand-punch programing language statements onto a deck of IBM cards, and put them in the card reader behind another deck of cards that was the compiler. The computer would then punch out an intermediate object file. Then you would have to put that object file into the card reader again behind a run-time deck, and then, more often than not, the computer would indicate an you left out a comma and you would have to do it all again. So, it took quite a long time. But it was a real computer. And it whizzed, clanked, and hummed. And the lights flashed. But most disquieting of all, wasn't I just taught that according to Descartes, the world was made out of two things—ideas and material substance—that didn't interact? But in this machine, the pattern of flashing lights represented ideas currently under the machine's attention. And physical action was exemplified by the console typewriter, whose keys moved and typed messages, but without any person or finger pressing them, as if by a ghost. Ideas and material substance did interact! It's almost impossible to convey how exciting this was. Modern computers, a billion times faster and much smaller, make this process almost invisible. But this computer made the presumed philosophically impossible visible.

Yost: So, timesharing would be an improvement.

Card: Yes. Yes. And then they got an IBM System 360 Model 30 computer that had a disk operating system on it and later on a Model 40. But we still to load the disk, we had to load trays of 2,000 cards per tray—and a lot of trays! I wanted to take a year off before going to the next stage—so I managed to get a job with the computer center. They made me the assistant to the director, then the director had to go work on his thesis for a while, so I sort of kept the place going while they looked for a permanent director. This was a tiny computer compared to the computers at University of Minnesota or other big schools at the time. But one thing we could do that other schools didn't do was: if a student could pass my test, which is basically I would turn off the various devices and see if they could figure out how to turn them back on and get the computer again going, we would let the students sign out the computer overnight and personally operate it for a number of hours. And so, in 1966, a student could have a personal computer—a personal mainframe! Most people who went into computer science at the time figured out a way to spend some intimate time like that with a computer. This was a way you could do it at Oberlin that you couldn't at University of Minnesota, and many students went on to graduate school in computer science from Oberlin despite the fact that the college had such small computers and no computer science department.

Yost: Yes. I have encountered that before in interviewing people. In 1967, you started at Carnegie Mellon as a graduate student, can you tell me about that—and that's in computer science, or is it?

Card: It's a peculiar story, so I'll tell you that. I was at the computer center, and Herb Simon [a founder of the business school at Carnegie Mellon University and later a winner of the Nobel Prize in economics] came to give a talk [at Oberlin]. There was this Tuesday Chapel in a cavernous building called Finney Chapel. And there were these people who handed out slips and we had to sign in. It was a graduation requirement. You had to go to at least 10 of these. But they were really good. Herb Simon came and talked about cognitive psychology and artificial intelligence and computer simulation and all this. And later, after the talk, my manager had arranged for half a dozen students to talk with Simon, and I was in that group. And I asked him how I could go and study this, how I could go to Carnegie Mellon University where Simon was (Carnegie Institute of Technology at the time). He said there was an interdisciplinary program called "Systems and Communications Sciences" and you could participate in various ways. Which department you were associated with didn't make much difference; it was this program that mattered. I applied instantly and I got in, but when I got there, they put me in the Psychology Department. I intended to be in their Computer Science Department. So, after the first semester, I went to see Professor Allen Newell (another founder of the program), and I said, "How come I signed up for this program, and it looks like they're just making me be a psychology PhD student?" And he said, "Well actually, the day you arrived, the program ended." So, he said, "We'll give you your choice, you can either do the program as it was, we'll grandfather you in, or you can be a psychology graduate student, or you could make your own program." Well, of course, nothing beats making your own program.

Yost: Right.

Card: So, to the astonishment of the faculty, who didn't know that students could rip up the requirements they were given and substitute a new set, I laid out a new program and jettisoned some of the qualifiers in psychology and substituted qualifiers in computer science. So, it was this sort of tailor-made program, and I stacked my committee with Newell, Simon and the chair of the psychology department so there was no power that could go against us whether the rest of the faculty approved or not.

Yost: Do you recall who the chair of psychology was at the time?

Card: Yes. Lee Gregg. Two G's.

Yost: What were you most interested in researching?

Card: Computer simulation of cognitive things. While I was there, I took a course in programing languages and the programing language that impressed me most was LISP. We didn't study LISP like 'Give me the manual', we spent a couple weeks on the lambda calculus, and then LISP sort of pops out. But it has the grounding in the lambda calculus, so it's not just another random programing language. You sort of have guarantees that you can do certain things.

Yost: In those early years, can you describe Allen Newell and Herbert Simon as mentors?

Card: Yes. Now I didn't have too much interaction with Simon, although I'll tell you one story. Simon used to go to colloquia. I was even the person who arranged the colloquia at one point. And he would sit there in the colloquium, looking like he was falling asleep but about 10 minutes before the end, right when the speaker had laid out all of his data and was about to give the results, Simon would raise his hand and he would predict the results. The speaker would typically sort of wilt a little bit. But I thought this is cool, just by knowing some things about cognitive structure and a few constants you could take a cognitive study and could calculate the answer to it. Simon had a number of constants that he used to do this. In my thesis, there is a chapter, which later became a chapter in a handbook called the Model Human Processor, and my goal of that was to predict everything Simon could and at least one thing that Simon couldn't.

Yost: And you completed your masters in 1970.

Card: The masters is no big deal in this thing, it's sort of, progress along the way.

Yost: I see that you also worked for a half decade as a consultant and statistical computer programmer at the Psychological Service of Pittsburgh. Can you describe that work?

Card: Yes. Various little jobs came my way as a graduate student. I don't know if I was good at getting them or something. Like I was the regional testing center for the travel agent certification board. So, if you were a travel agent, and you needed certification, you came and took this test, which I proctored. Another job was for this company called Psychological Service of Pittsburgh. They did psychological testing. They were under the gun from the government to show that their tests were valid because they were used to select people for employment. And so, they had all of this data processing they wanted done. Well this was just at the time when SPSS came out. And I knew about SPSS. So, I took in and processed the jobs with SPSS. And I was expensive. In fact, some faculty said I was making more per hour than they were. But Psychological Service tried to replace me with an IT student who was programing the thing in assembler language. And I was competing against him in SPSS. No contest. After that, they didn't complain anymore. So, this is no big deal. It was just that—as a part-time job, as a student—it was to help pay my bills.

Yost: In the early 1970s, you're doing various jobs part-time and working on your doctorate?

Card: Yes. I also got jobs as a research assistant for faculty and so on.

Yost: In 1974, you joined Xerox PARC as a scientist.

Card: Yes.

Yost: Can you tell me how that came about?

Card: Yes. Well, there's a story to that. In fact, Newell tells it—it's on videotape, on the web somewhere in his last lecture. He wanted to spend more time out in San Francisco because he's from San Francisco, and Pittsburgh wasn't as exciting at the time. And he was an early consultant for PARC. So, he convinced me and another graduate student who had interests in this area of cognitive science, that what we wanted to do was come out and work at PARC. Now, from our point of view, the information processing psychology that Newell and Simon had developed was ripe for some testing. This was an opportunity to go to the next step and see if it would really hold up. We had this notion that psychology really wasn't good enough, as good as it should be. We should be able to do more things with it. So, brash, young men that we were—

Yost: And, the other person was Tom Moran?

Card: Tom Moran. Yes. You needed two, you know, in order to make this work. So, the idea was to go off and create a kind of applied information processing psychology that you can use like you use chemistry and physics to predict stuff. It should be oriented towards prediction, zero parameter prediction. Just like you know, you want to find out what the torque on an engine in your car is, before you build it, you can make some predictions based on the size of the coil and all that sort of stuff. So, we thought we could do that. We thought it would force psychology to bend a different way. And that would help with some of the problems of using it. I mean, there's other problems with using it, it's just hard to do.

Yost: You wrote that there was sort of a mantra of "Nothing drives theory like a good, applied problem".

Card: Yes, a problem.

Yost: Can you expand upon that a little bit?

Card: We were full of bravado. If you look at the history of technology, a lot of times the—well, look at keeping time and determining longitude of ships. That drove a lot of things. Nothing drives basic research like a good, applied problem. Part of the purpose of that statement was, there was this recoil within psychology about doing applied things, because it was thought that people doing applied work were inferior. Outside of psychology, it was thought that psychology itself wasn't good enough to do it. That psychology was mushy and stuff. Which a fair amount of psychology was, but this slogan was our way of countering that.

Yost: Obviously, Allen Newell was a major influence on you. Any psychologists that you're reading at that point in time—

Card: Broadbent. Donald Broadbent.

Yost: That were particularly influential to you.

Card: Yes.

Yost: Can you discuss him?

Card: Yes. And Welford, A.T. Welford. Well, these guys were a part of an applied psychology unit at Cambridge, I think. Later, I can't think of the institute that Welford headed, but they set out to apply academic psychology to applied problems during World War II, and that's where a lot of results came from, so I thought they were in the spirit that we wanted.

Yost: In mathematical psychology, which really gets started with a few papers in the early 1950s, was that also in part an influence?

Card: Only…

Yost: Estes and some others.

Card: Yes. Only a little bit. I read some as an undergraduate at Oberlin. I discovered and read Atkinson, Bower, and Crothers, who are mathematical psychologists. This was a book on learning. In fact, I programed the algorithms in the book for the computer. I think I used the most computing time of any student programing and running those models. Remember, back then, the department used to have to pay for the computer time that its students and faculty used.

Yost: It was fairly expensive.

Card: Yes. Now, you know, you don't even think of it—it's free. Just buy the computer.

Yost: Can you tell me, in the mid-1970s, how the Applied Information Psychology Group was interacting with the rest of the research enterprise at Xerox PARC?

Card: Yes. That's interesting because we didn't—we were in the systems and communication sciences lab because, for other interesting reasons, because, that's where this prototype office of the future system resided. And we thought we were going to Xerox to study programing. I think it's good we didn't because I think that's a difficult task to study. So, we weren't formally part of any group; we ended up as our own funny little unit. At the micro-level, Tom and I took turns each month being the head of the group, at least for a while. And we were sort of in the orbit of a larger group run by Bill English. Bill English is the engineer who really made a lot of Engelbart's stuff happen.

Yost: Back in 1960, JCR Licklider published an important paper in which he wrote about interaction and symbiosis, and a future of ultra-intelligent machines. And a few years later, in 1962, he became founding director of the [ARPA] Information Processing Techniques Office [IPTO]. Did you see what you were doing at that time as working on an agenda in some ways that he had laid out?

Card: Yes, because that was really the Xerox PARC agenda. I mean, Licklider came to ARPA, sort of set that up. He had this vision of interactive computing, and he thought he had a chance to do it. He built the infrastructure for all this—the graduate students that got educated on DARPA contracts became the leaders in the field later. The whole reason the United States is in its dominant position in information technology, is because of ARPA. And Licklider were an important part of that. Licklider's deputy at ARPA, Bob Taylor, came out to head the Computer Science Lab at Xerox PARC. And Bob Taylor was sort of Butler's [Lampson, at PARC] boss. So, we were wired into Licklider's vision. Bob Taylor did Licklider's vision at PARC.

Yost: And in 1970, the funds from what was ARPA, when it became DARPA, had to be more defense-oriented, so that really kind of pushed Xerox PARC to take off where a lot of people that had been at universities and other places [funded prior by IPTO] wound up top people at Xerox PAR, right?.

Card.Right. Taylor moved to PARC, and he knew the top people in the ARPA community who were developing systems under Licklider's vision, since he had been funding them all. So, he just hired them and moved the best of the ARPA community to PARC. As a consequence, PARC was said to have—I don't know if this is literally true—but it was said to have 50 of the top hundred computer scientists in the world at one point. It was 10 years ahead in computing. The computing that happened in the Homebrew Computer Club that met at HP has gotten a lot of historical note, some PARC people went over there sometimes. That was interesting sociologically, but computer-wise, their computers and ideas of computers were 10 years behind what we had running.

Yost: How much of Allen Newell's time was he spending at Xerox PARC versus back in Pittsburgh at Carnegie Mellon?

Card: Quite a bit. He came out every month. He'd spend like a week a month at PARC. He had a condo on Second Street in Los Altos. The deal that he negotiated with PARC was that for 10 years, PARC would support us and let us just do whatever we wanted.

Yost: Exciting!

Card: Then at the end of 10 years, we would deliver a useful applied information processing psychology science to them. That's a typical Newell bargain. So, it's exciting until you look over the cliff of the canyon whose crumbling edge you're standing on. But we had a certain amount of attitude, so we loved that!

Yost: Can you talk about Tom Moran a bit as a collaborator?

Card: Tom Moran and I met in graduate school, of course, as Newell students. We even shared an office for a while there. Tom had been trained as an architect, and so he had a more developed sense of design than I did. He did a thesis that was remarkable. It was a psychological thesis, but it only had one subject, I think. Tom was responsible for a lot of insights. We worked fairly well together. One of the minor things he did, but that had a profound influence on me, was he wrote this data analysis program that took the data from keystrokes and mouse-movements and analyzed and drew diagrams of their chronometric relationships. This program was fundamental to our protocol analysis studies, where we tried to take apart user behavior. I got so many insights from this program.

Yost: So, I asked you about Licklider, I'm wondering if, even before that, work in cybernetics influenced you, say work by Norbert Weiner and McCullough and Pitts?

Card: Yes, I'd read those.

Yost: Man-machine interaction and symbiosis.

Card: Yes.

Yost: If that too was influential.

Card: Yes, well I was on this National Academy of Sciences study by Al Chapanis. The issue was that in psychology, people do experiments, but what methods have been developed that are computational, where you don't have to do an experiment to find the answer. There was a report that this study put out. We identified 40 non-experimental methods, but we could find only two of them that were ever validated anywhere or taught anywhere. One of these was timeline analysis for doing cockpit designs. The way people who would do cockpit designs is they would have a set of operations that would describe what people did and then they would use these to analyze especially complicated scenarios, for example, landing or defending yourself in the sky, as a way of looking for problems. In one scenario, carried out for 40 hours, you're running a bomber. So you predict, at half-second intervals, everything you do for 40 hours. I was interested in this literature, which was invisible to most people because there's a government clearing house and you have to be a contractor in order to get to the literature. So anyway, back to this study. This study really made an impression on me then. Imagine, the methods were very little validated and very little taught, and yet important design activities depended on them. Talk about an engineering discipline having sandy foundations! And so that was one of my inspirations. But I also—this is sort of an amusing story. Newell was asked to be on this National Academy of Sciences summer study in Woods Hole, Massachusetts on automation for combat fighter planes and he didn't want to do it, or he was too busy, so he proposed me for it. So, I arrived at the study, and there are all these eminent people, like Licklider was on this study, as was Matt Parsons, a famous person in human factors. Then there was me. It gradually became clear to me that I was actually the chair of one of the three panels of the study. They would've made Newell the chair, but Newell made me the chair. One panel was battle tactics. Another panel was technical subsystems (the mechanical parts, the electronics and everything). And the third pane was us, human factors. As we proceeded, I noticed that the chair of the subsystems panel wasn't paying attention to the briefings. He was busy giving assignments to the members of his committee and starting to organize the briefing and report his committee would have to produce at the end of the study. I was supposed to be his peer, but I also noticed that he was the Chief Scientist of the Air Force, whereas I was a graduate student! So presumably (unlike. me) he was very experienced in such studies. So, I took my cues from him, and I issued instructions and made assignments like I saw him do. That's how I learned to run a study. Licklider was on my panel, so he worked for me, sort of.

Yost: Wow! That must've been quite an experience!

Card: Yes! Licklider was especially interesting, because during the meeting, as people were talking, he would write this perfect essay on the topic under discussion. Like, one of the briefers said they have this system in which they could automatically fly the plane around in a dog fight. Licklider raised his hand and said, "Oh that means you're too predictable. All they'd have to do is know your algorithm, and they can shoot you down." The guy had never considered this before. This study put me in touch with a pile of literature, some of which I have still. Hand-drawn figures about searching for items on a display, and that sort of stuff. Problem is this literature never gets polished enough to get published. I think Matt Parsons and somebody else rescued part of it and made a book of it. But there's a lot of stuff that's not in the book. So, as I was on this committee, I said, "I have to talk to real pilots in order to understand this stuff. And I have to see it." Because I, of course, wasn't working full-time in the aircraft industry. There was a national guard airbase not too far away from Woods Hole where the study was. They flew in an F-16 for me, and then let me climb up into the cockpit and turned on all the displays while the plane was on the ground, so I could see what they were like. And then they flew in a F-18. There were only 20 F-18s in the world at the time. This marine test pilot was testing it, so they just diverted him up to the Woods Hole so that I could sit in the cockpit and see what it was like. They also flew in three pilots to talk to me, one the chief test pilot of the Air Force and two combat pilots.

Yost: Fascinating. What timespan was this committee leadership and work?

Card: Around 1982.

Yost: Ok.

Card: I learned a lot from this panel. I had to do a brief-out in front of 30 generals—Air Force generals are not people known for taking nonsense kindly. But because I'd thought of it, mine was the only set of slides with a font big enough so you could read the slide. So, I actually did pretty well.

Yost: Back in I believe it was 1967, the Human Factors and Ergonomics Society forms. Do you recall when you first became involved with that Human Factors Society?

Card: Well, I never really became too involved. I always subscribed to their general journal. I tried to join one of their organizations, but somehow, bureaucratically, they never answered my requests. I read some of that literature, but at the time it was more about knobs and dials, perceptual things and motor things, and we wanted cognitive things. So, we—you could say that we were filling in part of the spaces that hadn't been filled in yet. But also, it was, they would run experiments. A lot of people who got into human factors were psychologists, because a psychologist knows how to run a controlled experiment. But we wanted to calculate stuff.

Yost: In 1979, Don Norman, Roger Shank and Allan Collins formed the Cognitive Science Society—

Card: I was at the first meeting.

Yost: You were at the first meeting. Can you describe that meeting and that society in its early years?

Card: Yes, well there was—at the first meeting was where, I guess, the situation was set. So, it was supposed to be psychology, cognitive psychology, linguistics, AI and so on. But the groups were different sizes or power levels or something. I remember some discussion about that—I should be careful of what I say because I don't remember which group it was that objected; it might have been the psychologists. They were talking about setting up conferences and organizations and so on. One group, I think the psychologists, were afraid they'd be dominated by the AI people. And the AI people wanted to set it up a certain way. They were doing very well with having professionals running their conference, and then just paying for professionals to set up stuff. I don't think the psychologists wanted that. Don't quote me on exactly what that was.

Yost: Ok.

Card: But the psychologists and the AI people sort of broke away. So instead of the AI conference being the cognitive science conference, they were going to be separate, and that sort of doomed the highest aspirations for them. I mean the Cognitive Science Society is still around, but it didn't grow to the power that the AI is now.

Yost: Newell had the insight that behavior can be approximated or approximately indexed by the log of time to performing unit of behavior, from there, grouped into biological, cognitive, rational and social behavior bands. Can you describe how this insight factored into the research program of Allen Newell, yourself and Tom Moran and what you did at Xerox PARC in that area?

Card: Well, Newell—we asked Newell to give a talk, maybe it was 10 years in—somewhere around there, to the CHI conference. So, the CHI conference got started, and it gradually started to bloom a little bit. And we asked Newell to give the plenary, and when you sit Newell off in a corner with an assignment like that, something brilliant always comes out of it. And the time scale was his insight, one of his insights for this.

Yost: Would that have been in Boston at the first SIGCHI—

Card: It wasn't the first SIGCHI.

Yost: Ok.

Card: But it was—

Yost: Early on.

Card: It was early on. And when he did that, that just sort of made stuff fall into place because you could, I mean, it's amazing that time will do this, but by just sort of spreading it out like that, you could have a way of guessing what—where something belonged in all this just on the basis of the time. Simon, in his book, *Sciences of the Artificial*, which is a cool book, had republished an essay in the back on nearly-decomposable systems. And this says that any system, even in nature, is going to have a logarithmic structure, and it's going to be—the parts are going to be almost decomposable into component parts—not independent but—you're going to be able to identify them. But not quite. So there there's a little leakiness that goes over on one to the other. But there's another idea that comes out of this—Simon said that things are decomposable in this way, and if they are linear and everything, then you don't need to know the mechanisms for it. This is the lesser-appreciated part of Simon's insight into bounded rationality. So, at Carnegie Mellon, everybody thought that bounded rationality was a limitation on what the economists could do, because the economists assume rationality for everything. But you can show that you couldn't nearly compute what you need to in order to make the rational action happen. So, you have to have heuristics. Bounded rationality was usually trotted out like that is that, it's a limitation on rational systems. But I saw it as a limitation that spoke to the psychologists too, because, if a system <u>is</u> rational that is, if you can predict what it would do by computing the payoffs for each possible action, then optimizing by choosing the action with the maximum payoff, you don't, basically, need to know any psychology in order to predict behavior. Take short-term memory. You have a telephone number that somebody gives you, and it's too large for you to remember. You need to know some psychology to figure out how large a number you can have before you start forgetting parts of it. Looking at it from this other point of view, bounded rationality, it says, well, obviously you just switch the tasks so you can do it in a rational way, so you write it down the number on a piece of paper. And now you can do the task. which is someone tells you a telephone number and you recall it because you've made it this a rational task again. You don't need to know about the internal implementation of the task. So, you don't need psychology to do psychology. As Simon puts it, the complexity of behavior is not in the behaving organism, but in the environment. And later on, John Anderson wrote a book on this "rational analysis". So, he agreed with that insight. In all of these cases you want to see if you can't patch things up so that there's this rational set of operators that you can use to get the goal that you want

Yost: So, in 1978, you complete and defend your dissertation. Can you describe your dissertation work?

Card: Yes. I essentially did two dissertations. When I was a student back at Carnegie Mellon, I stopped into Newell's office one day and he assigned me a dissertation topic. I was supposed to take a field, concept learning psychology, which is one of those little fields in psychology that had this and that empirical results, and. I was supposed to create an AI machine that "understood concept learning". That could design, run, and analyze experiments to move this field of psychology forward without a psychologist. That was hard, because it was difficult to represent the semantics involved and have everything fit together. And remember, we didn't have all that much compute power. Nor did we have a theory for how to do that. So I struggled on that thesis. And I came to PARC before I finished the thesis, so that I would still have my job, and still have a wife, who was already out here. At PARC, I completed this dissertation and sent it over to Newell who marked it up and I was ready to defend it. I don't know if it was Simon's or it was Newell's idea or mine, but in the course of doing my work at PARC, I'd written all these papers with Tom Moran and Al Newell which had been accepted. We decided that I could just put the papers together and make a dissertation out of that, and that second dissertation would be better than the one I had originally written.

Yost: And that was kind of the broader HCI, psychology of HCI?

Card: Yes. Well it was these isolated papers at the start, but I developed a plan to fit them into a structured framework. So the ironic thing is that when I came out to give my job talk for PARC, I presented the older dissertation because, of course, I hadn't done the other work yet, and I got hired on the basis of the older dissertation. The new dissertation was called "Studies in the Psychology of Computer Text Editing".

Yost: In the early years at Xerox PARC, can you tell me to what degree there was interaction with the industry, the laboratory, and the broader research community, and especially out here in Silicon Valley, but also nationally., You mentioned Bill English and Engelbart, connections with SRI, with Stanford, with other IT firms.

Card: Yes. Well, all that stuff was connected.

Yost: So, I'd asked about connections in the area and interactions of scientists between institutions.

Card: Some PARC people went over to the Homebrew Computer Club meetings which I think met at HP. There were a lot of connections with various academic departments. The computer science laboratories at PARC were heavily recruited from Carnegie Mellon, MIT, and Stanford. The same people I saw in the halls at PARC were the same people I'd seen in the halls at Carnegie Mellon or MIT. Faculty at those places were also PARC consultants. Most people seemed to have DARPA connections, like people from SRI, Utah, BBN, or Cal Tech. Although PARC got the reputation for being a black hole for research at first, they didn't intend to be. Xerox didn't do too much patenting, possibly because it had a consent decree that said it had to publish its patents so everybody else could see them. Around this time, I had quite a bit of interaction with the government. Since I apparently did well on the National Academy study with the airplanes, I ended up on the National Research Council Board on Army Science and Technology. And I let that happen, because Xerox is a big organization and I wanted to understand that, and how you understand it is you look at other organizations that are even magnitudes larger, which is the Army, and so I went with my committee all through how the Army developed new systems, and that was quite an education, so I could see the parts that were like Xerox but also different. I could get into notions of scale and stuff and get a sense of different systems that had human components to them. Here are some other stories about connection at the time. Several of them center on interesting interactions that I had with Licklider. I had my scheme for calculating behavior, my Model Human Processor, and I was telling Licklider about this at Woods Hole. I was just writing it down at the time, and he said that's wrong. He said, that for one parameter I had in it, the motor parameter (which gave the time of the fastest motor action), should be 700 ms, not 0.1 sec as in my model. A tenth of a second was very nice, because it was the same value for several parameters. But Licklider said it should be 700 ms. So, I argued. I said, "No, it can't be, I read this whole literature". Of course, Licklider invented half the literature I was reading. So, it was a little bit audacious to insist I was right. He said "All right. Let's find out." We were at breakfast in the National Academy of Sciences mansion that they have there in Woods Hole. We started measuring this constant I had used called the tapping rate. I was the tapper, and Licklider held the watch to measure time, and somebody else counted the strokes. We did this whole experiment, banging spoons on the table as fast as possible. The rest of the people out there on the dining porch, thought we were nuts, I'm sure, because we went pounding on the table "Bang! Bang! Bang!" Licklider was right, of course. I had to revise my whole model -- but then it fit better than before!

Yost: Can you talk about the application of the Fitts's Law theory; of the mouse and how that impacted your research. I understand it was introduced on the Xerox Star in 1982?

Card: Yes. Well, it's interesting because it illustrates the thing we were trying to do. I ended up in Bill English's group or English's orbit. He had been the one who did the engineering on the mouse for Engelbart. He thought that there were devices better than the mouse. He wanted to test some of them. And he didn't have very much time, so would I help him on the test? I set up a human factors test with four devices. Here's what a mouse looked like [picking up an historical mouse]: two wheels at a perpendicular angle. The original SRI versions of the mouse were huge. This version has been miniaturized to make it easier to hold,

Yost: Ok.

Card: I might not be showing you anything you don't already know.

Yost: I haven't seen one. I have seen pictures, but not in person.

Card: Well, this one has colored buttons.

Yost: Buttons.

Card: Here's an Alto manual. This has a manual for the Bravo editor in it. Bravo is a text editing application that uses the mouse, and which eventually became Microsoft Word. They have color-coded the keys on the mouse so that the documentation of the text editor can refer to the buttons by color in instead of by function.

Yost: Very interesting. So, 1982…

Card: Well, I wanted to tell you the story of the mouse.

Yost: Oh, sure.

Card: I made off with some of these old things because they weren't functional for the current systems, and they would just be destroyed. As I said, this mouse works even though you have to move it with the wheels sliding sideways. It's remarkable that it works. English was interested in comparing the mouse against these other devices. I did the experiments, and the mouse was fastest by a lot. But then in connection with what I said we were going to do in our research, I did a model of each device, and that told me why each one took the time it did to select a target—what was sensitive about its construction and its parameters. Then A.T. Welford came and visited me. I think he wanted a job for one of his students. He came and visited, and I was telling him about this two-tenths of a second parameter that I'd sort of condensed from the literature. He said, "Actually, it's not two-tenths of a second. It's one-tenth of a second, that's the magic number." So that was useful. Then I discussed the mouse with him. And he was saying, "Try Fitts's Law." Fitts's Law says that the time it takes to move something from some initial position to a target position is proportionate to the base 2 logarithm of the ratio of the distance to the target to the size of the target, plus a little bit to keep bad things from happening to the law at the low end of the parameters. And there was one Saturday, I think, I was sitting in my office trying these things, and all of the points, when I plotted movement time against this expression which is now called the 'Index of Difficulty,' the points lined up in a straight line. The points running in a straight line meant that the mouse followed Fitts's law. But what was more interesting was that the Index of Difficulty, which is the slope of the straight line, was about 10 bits per second. And what was interesting about that is that that's the slope that you get if you just take the hand without the mouse and just move it to the target. What that meant was that the mouse device wasn't the limiting factor; the speed of the eye/hand coordination system of the human was the limiting factor. And what that meant was that if you put a device on the market based on the mouse, nobody would be able to beat it, because it was already at the optimum. So, I had this succession of realizations, sort of about one every 15 minutes, as I rummaged through the data, and that was really exciting because that it was an example of what we said we were going to do. There's been some dispute about the exact numbers, because it's sensitive to the details of exactly how you set up the experiment, and how you handle errors, and whether you force the line to go through the origin, but the basic result still holds.

Yost: That must've been great to get to that result.

Card: I gave a talk on this in the PARC Computer Science Laboratory and Butler Lampson wasn't there that day, but I guess Butler heard about my results and he had a different result, a different way of thinking about it. So, he came down to beat me up, but fortunately I guess I was gone that day.

Yost: Can you tell me about your research on—

Card: I want to expand—

Yost: Sure.

Card: Expand on that little bit on the mouse, because it gets to another thing here. So, there are several things we get out of theory, and we can illustrate this with the mouse. One is we get to understand why the empirical finding is the way it is. And when you can find mechanisms like that, it's worth as much as the statistics of counterbalancing and everything, because there's a reason and a constraint for the results. The empirical reason why the mouse performs at 10 bits per second is because the mouse is moved with the wrist and measurement of motor movements show that the wrist produces a Fitts's Law slope of 10 bits per second. Inventing new artifacts is another way we can use theory. Given that we now understand how the wrist-moving mouse works, we can back-project from the Fitts's Law theory and invent novel designs and predict their performance before we even build the new device. If we can set up a device that yields a Fitts's Law slope of faster than 10 bits per second, we can beat the mouse! Other research from the literature shows that you can get a Fitts's Law slope of 42 bits per second by using the fingers. The theory says we can beat the mouse, if we can find a way to operate a pointing device with the fingers. This suggests a device shaped like a pencil, stuck on some weighted base to hold the pencil upward and at an angle--like a pencil in use--and to make it easy to grasp. The opportunity to build this faster mouse came when Xerox designed the second version of the Xerox Star. The designer Bill Muggeridge [founder] of IDEO came up to PARC with a whole crate of mouse designs; he had just sort of been wandering around in this design space and wanted some help. And so, I told him about my theory of the mouse and showed him the pencil-stuck-in-the-eraser conceptual prototype. Bill and his associates then went away and designed a family of “finger mouse” devices using our principles that looked a lot better than a pencil stuck in an eraser. This showed that by using the theory, we could do what we said in the first place was impossible; we could beat the speed of the mouse, and we could do that by changing the muscles that controlled the device. Alas, we couldn’t get Xerox to fund building the finger mouse designs because they said, the mouse was already better than anyone else’s pointing device, so why should they spend money to improve it? Conversely, we can use theory to reject candidate device designs without even building prototypes. There was this thing called the “headmouse” at the time. You wore it on your head and wherever you turned your head, it moved the cursor. But the muscles moving it were the neck. The neck yields a Fitts’s Law slope of 5 bits per second. So, I said the headmouse would never win, and one of the engineers said, “Oh, you just have more experience with the mouse, and if you use this enough, it’ll be as fast as the mouse.” So, I gave him a headmouse, and then one of my greatest scientific satisfactions ever was that over the next week I saw him walking around with a stiff neck, because, of course, he didn’t beat the mouse--righteous punishment visited on him by the gods for the folly of siding with mere intuition over my theoretical predictions. I should mention one more way in which theory helps experiment, important in industry. The original experiment that we did took six months to program and run. It simulated in some detail the text editing task. But since we know now that the device is governed by Fitts’s Law, we just have to measure the slope of Fitts’s Law while using the device. We know what details <u>we can leave out</u>. For example, targets don’t even have to be text; they could just be rectangles—or even just lines. Fitts’s Law doesn’t care. That’s a much easier experiment. You just have to have two targets and you move the mouse between them as fast as you can. So, assessing the speed of the mouse, which originally took six months could now be done in ten minutes--tremendous efficiency gains for measuring a device’s performance because of theory.

Yost: Yes, it completely—

Card: Tremendous efficiency for figuring out the details.

Yost: It changes things.

Card: Shumin Zhai did some experiments which extended the predictions of how the mouse would move in more complicated tasks. He derived and generalized the theory, and eventually invented the "gesture keyboard". You move your finger around the different keys without removing your finger from the display, and this can be faster than typing. A really skilled person can input text at almost 100 words per minute, I think. But all these little things that are like Fitts's Law fit together now in this common framework, and you can use pieces of them together. That's sort of what I wanted to happen… all these little pieces start fitting together. And that's how you build an HCI discipline.

Yost: Can you tell about how on the mouse, the research science and engineering side in creating one is interacting with developing products, developing a product that can be manufactured in volume?

Card: I had some experiences with that. Well, let's take the first; deciding to introduce the mouse. There was a part of Xerox that was dead set against using the mouse, because it had a chord that tangled everything and hung off a box instead of being embedded in the device. They didn't like parts hanging off the device. So, I went down to Los Angeles and I gave my empirical results, and the room was hostile--didn't I forget this or that? But then I got to the theoretical part of why the results had to be the way they were. They just sat there in sullen silence and glared, because there was nothing they could say. And sure enough, we introduced the mouse, and nobody ever beat it. So, what we predicted was true. So, what was the question?

Yost: Turning the R&D of one into a manufactured product.

Card: Yes, well, so the mouse was a manufactured product that came out of this. In fact, Apple, asked for my results before they took the risk of going to a mouse. Nobody thinks of Apple before mice, but they had to go through this as well because the mouse was a very unconventional device. By understanding why the mouse worked, it de-riskified a radical product performance-wise and market-wise, in the sense that no one was likely to build a faster product, and if they did, we knew how to beat them Another use that we did with the theory was that, they [Xerox] thought they were having some problem with the mouse electronic circuits on the Xerox Star, and they weren't quite sure what it was. The question was, were the circuits fast enough to handle the peak movement speed of the mouse? The mouse cursor has got to track the mouse movement of the hand and that requires circuits; and faster circuits are more expensive. The product was already over cost. This problem came up during lunch in the PARC cafeteria with some product engineers from the Xerox product organization, I think Bill Verplank brought i up. I did a little derivation; it said the top speed of the mouse would be something like 4.5 times some constant times the distance of its movement. And so, if you take the longest distance you can move the mouse--the diagonal of the display—the top speed of the cursor, by our study, would need to travel too fast for the existing circuits according to my calculation. After lunch, we went down to the lab and empirically verified this, so we had a theoretical result and had empirically showed that the theoretical result was true, and that they had to fix the computer and spend more on the circuits. In an afternoon, we had derived a realistic model, analyzed it, and answered the question. We had found and fixed the problem, and just as important, we had done this without letting the schedule slip and for very low manpower cost. I could give you lots of empirical, or commercial results of these things, but I don't know when you want me to stop. I'll give one more quick example. We were trying to sell the Xerox optical mouse to Microsoft. This mouse used a clever optical sensor which was much smaller than the mechanical assembly of previous mice. This let the sensor be moved to the front of the mouse in such a way as to shift more of the control of the mouse. It wasn't a full fingermouse, but it definitely was faster and more adept. Steve Balmer, president of Microsoft, came to PARC and got a demo. I don't think in the end Xerox became Microsoft's mouse supplier, but Microsoft was convinced to follow Xerox's mouse design rules, and they moved their sensor from the back of the mouse to the front, where it could exploit the fingers. Again, we were able to use the mouse theory to advance an industrial product. Here's a slide to summarize an overview of our results [shows an illustration containing theories, technologies]. And then here's the sort of mid-level phenomenon that describes what device components relate to what theories and then up here are products the components go into. So... I don't see the mouse one right on here. This is too small anyway. Anyway, these are actual products that were put out and I identified at least a dozen products that came out of the research. This is my trying to make the case that the core research is part of a stream that comes out with products in the end. I tried to keep track of this to see the extent to which we can get theories in this area that we can use for commercial use. As another way to identify commercial value, I have 50 patents that are theory generated. The point is to challenge the theories to be practical enough so that you can build things with them. The fastest product transfer I ever did was with Larry Tessler from Apple. He wandered by and I was showing him how I had inserted this line between groups of items that helps with visual search and visual learning of the menus. Nowadays, it's a standard part of pull-down menus.

Yost: Yes.

Card: I was showing him this and why it worked, and so he went back into Apple and that afternoon, they put it in an Apple product. Other people imitated it, and now it's universal.

Yost: And was that for the Lisa?

Card: Yes. I think so. So, technology transfer took like half an hour. I'll give one other example, Rooms. We noticed that even though windows were really great on workstations, they were a disaster on PCs, which for cost reasons had much smaller, lower resolution display screens. We wanted to figure out why they worked for one and not the other. So we put in a measuring device that would allow us to track window usage, and what turned out can be explained by reference to a surprisingly similar situation in virtual memory operating systems. In a computer program, if you plot how often different memory locations are accessed in between accessing the same memory location again, you get a distribution called the 'locality of reference'. Memory locations are not uniformly accessed. They are accessed in bunches, and this is the basis of virtual memory operating systems—you can pretend your computer has a really large memory as long as you don't get into a situation where you need a large block of memory referenced together which is larger than the physical memory you actually have on hand. So, if you have a clump of things that you are touching together, and that clump is too big to fit in physical memory, then you know your virtual memory system goes to hell and 99 percent of the effort goes into swapping the contents of memory in and out. The same logic applies to windows. If the virtual screen, that is, the amount of area taken to make everything visible, is large relative to the physical area of the screen, then you'll spend most of your time moving around among the windows. This is a non-linear process so you can be going ok, and then you get some big windows and it goes to hell. So, here's the phenomenon, and now we have the theory of the phenomenon, right before, now how do we back-project this theory to get a new gizmo out of it? We back-project this by realizing that the working set theory and virtual memory operating systems are formally a lot like the windows problem that we were having and so we take one of the policies we developed for operating systems, in this case, a preloading policy that says, I can sense when I'm getting into a place where I'm going to use this set of references, so I swap them all in at once, and then when I'm going to need some other ones, I swap those all in at once instead of sequentially. This gives me several contexts to work in, all of which correspond pretty closely to a task. So maybe I'm doing my email, and I'm going to use this one set of things. Maybe I'm also working on Project A, and I'm going to use a different set of things, Project B. I worked on this with Austin Henderson. Austin was a genius at conceptualizing and building abstractions. We divided the screen into a number of work areas we called Rooms, and we had to change the notion of what a window is a little bit. Austin invented "placements". Placements carry the parameters that keep track of the height, width, shape, and x,y location of the left bottom corner of a window. Windows are redefined to be their usual definition less the parameters that have been moved into placements. This repartitioning allows the same window to seem to be in more than one Room at the same time. It could be big or small, or in a different position depending on which Room it was in. This was equivalent to allowing the same window to be used for different tasks. The user had an overview of all Rooms and could zoom into any one, work inside that one Room (set of windows), and then zoom into another Room and work on another set of windows. Rooms became about five different products, launched by different Xerox companies. It won the Runner-Up Prize at the national COMDEX trade show for Best in Show. Several companies licensed it for their machines.

Yost: And was that the only option, to license, given the Consent Decree?

Card: Well, this is a—

Yost: Or this is later—?

Card: We were far away from the Consent Decree.

Yost: After that no longer applied.

Card: So, this fills in my steps like I was doing before. Step 1. We form a theoretical analysis of the core of the problem. In this case, window problems are characterized by working set analyses taken from virtual memory operating systems. Step 2. We back-project solutions of this old problem onto the current problem. In this case, we back-project a pre-loading policy for virtual-memory operating systems by inventing the Rooms system of virtual window sets. Superficially, windows and virtual memory operating systems don't look anything like each other. That's why this is such a rich technique. We can pull deep analogies from other areas that wouldn't be obvious in the original area.

Yost: In showing me that image or slide, you stated how management didn't like it. Is it because they wanted to contain basic research expense and they didn't see, they didn't understand, what was actually created?

Card: No. It's that he knew if he challenged me on any one of these boxes, he was in for a 20-minute lecture. And after a few of those… But he really couldn't argue that they weren't true. So, he was annoyed that I got to do what I wanted to. . . I want to add that Mark Bernstein, who was a physicist, was very supportive of me and my group, and I'm very grateful for it, but I had to earn it. We were essentially trying to be helpful to his goal of making PARC pay-off commercially. I was always evaluated by physicists or computer scientists or linguists. They're not known for being sympathetic to psychology. But, remember, we picked such an environment on purpose for just that reason.

Yost: So, you were at the 1982 Gaithersburg meeting?

Card: Yes.

Yost: Can you describe that and were you at the evening discussion group where SIGCHI was founded?

Card: Yes. I was, because we were part of the founders.

Yost: What was the significance of that to you, the participation in helping to found a new discipline/specialty called Human Computer Interaction, HCI?

Card: Well that seems like it was right in line with our original goals, and so there was a lot of energy in people willing to do stuff. For researchers, having to put out our research, we didn't have time to do everything ourselves. And there are other people who are better organizers than we were, so if they could be recruited to do this, that was wonderful. I mean, things would maybe be different than what they would be if did it all ourselves. But probably better than if we had to do it all ourselves.

Yost: I understand from Don Norman, who was at that evening discussion that there was some debate about where it would reside, the academic society, and some favored ACM. Don Norman argued against ACM, but Ben Schneiderman argued for ACM.

Card: What did he argue for?

Yost: He [Don] just had some reasons why he thought it, I think, would succeed more as an interdisciplinary field if it were not ACM.

Card: I see.

Card: So, independent.

Yost: Either independent, or possibly Cognitive Science, I think.

Card: But I don't think that argument would have been beneficial in the long run because in order for this area to really get going, it had to have university professors who would teach it. They had to have textbooks and everything because the content of the field had to be defined. If every school treated it as a different field, then it wouldn't have the coherence so they could build up a strong field. And professors had to get tenure, and ACM, computer science professors, were unlikely to get tenure if not there [within ACM], more likely to get tenure if there was an ACM journal--

Yost: Right.

Card: --that had part of the literature in it. I can't image a group of computer science professors sitting around giving tenure to people who published in psychology.

Yost: That's an excellent point. And SIGCHI in its first few years had a number of people from human factors and practitioners [applied rather than research side of industry] and it quickly evolved to be more of an academic organization.

Card: Yes. Well, in fact, you know what I did? In 1986, I was the program chair for CHI and that was still when the program chair was chair of everything, not just the papers. So, I rewrote the Call for Papers. I didn't ask anybody, I just rewrote it, emphasizing computer science, and de-emphasizing cognitive science and psychology, even though cognitive science was sort of my dog in the ring. I thought that an area like this is not going to be successful unless it has the computer science in the base for it. And that call stayed untouched for about five years, and the composition of the submitted papers consequently changed more in the direction of computer science. About this time, I was appointed to an ACM HCI Curriculum Committee, which was charged with creating a reference curriculum for HCI. The academic members of the Committee had differing ideas of what should be in this curriculum, reflecting the many ways HCI was located and taught within their universities. At first, the Committee had difficulty making progress. I made two contributions that ended up moving things forward. First, I suggested defining HCI as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." This was parallel to the Newell, Perlis, and Haberman definition of computer science as "the study of computers and the phenomena surrounding them." The strategy was that a new discipline needs room to grow and flexible boundaries and this definition side-stepped some of the arguments we were having. The second suggestion was to write down all the teachable results that had accumulated for HCI and use these to generate courses, following a method Mary Shaw had used to generate the Carnegie Mellon University undergraduate computer science curriculum. To persuade the Committee this could work, I actually went off by myself for an afternoon and generated a draft of HCI teachable results just out of my head.

Yost: Certainly, the institutional resources in computer science are much greater—

Card: Oh, yes.

Yost: You mentioned the field needed early textbooks and your textbook, *The Psychology of Human Computer Interaction*, with Allen Newell and Tom Moran was and is a classic. Can you tell me about the writing of that book and how you structured the project, what your goals were and with what target audience?

Card: Yes. We wanted to kind of announce this area, or this attitude, really, of doing stuff and didn't have examples. That book started with the studies we had already published. And then it had an introductory chapter whose first draft Newell wrote. Newell has these wonderful turns of phrase. He wrote the part that said, "The human-computer interface is easy to find in a gross way—just follow a data path outward from the computer's central processor until you stumble across a human being." He helped put into the book the attitude that design is where the action is. Not evaluation or anything. And, let's see, we just passed the chapters around until, until they made sense, I guess. I remember each person would have their tasks to do, and then we'd get together for these long meetings where we'd go over everything page by page. I learned a lot about publishing during that.

Yost: Was your audience primarily thought to be students and early researchers in this field or were you also trying to gain a broader audience with that book?

Card: Well, there weren't very many people in that field, so that wasn't a great market. It was people who wanted to be in that field, I think. I mean, we've always had this mission to try and change the way psychology is done but…let's see, what else can I tell you about this? Because we had Newell, our publisher was very good to us. They were enamored of the fact that we could submit things in typeface, so we had to do all the type fonts and layouts and stuff. That took a lot of work, but it also meant that we had a lot of control. It also meant that we had the ultimate responsibility for keeping the manuscript safe. We had the book set up, so it went to seven levels of back-up copies. We created a new version with each save. We would all back it up on our local disks that would automagically send a copy over to another building and back it up on another server. And eventually we would print copies in case all of the back-ups were destroyed. We went all the way down to the lowest level of back-up on a couple of occasions. Things weren't very reliable.

Yost: So that was really necessary back at that time?

Card: Yes.

Yost: From Gaithersburg and the founding of CHI, the next meeting, the first, is in Boston. Are you going regularly in the early years?

Card: Yes. I didn't miss any of them until just recently.

Yost: In those early years, what do you see as the greatest successes of the society and what were the largest challenges or other paths that might have been taken?

Card: Well, the biggest successes, I think, were first the fact that you could get all these people to work together from the different disciplines. That was hard. So, the computer scientists eventually came in and, they were different from the psychologists. And then we—it was hard to get designers in, and we thought you had to get designers in. If you have any really big important group left outside, then the action could happen over there. And so, we had to capture where the action was. We—I remember sitting around at one program meeting, bemoaning the fact that there weren't any designer papers at the conference. I think I was the one who gently pointed out that there would've been, but we had just finished rejecting all of their submissions, because designers tend to write these very romantic papers. Like I remember one on the application of grid layout techniques to user interfaces justifying the design concept in terms of contributions to world peace. But we wanted the participation of the designers, because they had such good ideas, and we wanted design ideas to engage with cognitive ideas and computer science ideas and social ideas. But it was hard to keep these communities from flying apart. The only solution was to give the designers themselves responsibility for accepting design submissions, which they broadened beyond papers. This worked so well that they invented the term “design thinking” to show that they had actually been the ones who had invented the field all along. But that’s just what I wanted. Now the designers were engaged with the other disciplinary groups in a dialectic argument to the improvement of HCI. Another issue was that for many years, people would go to CHI and they’d say that there’s nothing new. It’s just all the same. But that’s not the best way of looking at it. There’s a phenomenon called the “dominant design”. Have you heard of this?

Yost: No, I haven’t, or perhaps in a different context, please elaborate.

Card: Interfaces are a part of technology, so I had to learn something about technology, and in particular, I had to unlearn what I had learned from reading Vannevar Bush. I originally thought that research and development was a sequential process that went from basic research to applied research to development. That’s not how technology works at all. So, if technology is a way of getting some principle of nature to do something you want, it often has parts that come together one by one. Take the typewriter. The typewriter developed over a number of years. In early models, when you hit the key, you couldn’t see what you typed, because the print heads were in the place where the heads hit the paper. In addition, you had many characters in the repertoire to be typed. If you wanted to accommodate both upper case and lower case, too, that’s lots more than the number of fingers you have. How are you going to select among a hundred and two characters with ten fingers? So, typewriter companies came up with various kinds of shifts. And for the classic Underwood 102 typewriter, there’s this thing that goes up and carries a ribbon and then the hammer hits and then it goes down. So, after you’re through typing the key, you can read what you just printed. So, where was I?

Yost: You had started on ‘dominant design’.

Card: Yes. Yes. This is very important. Progress comes about gradually as these inventions are added to new models of typewriters. Then finally you've accumulated enough inventions that a product is finally introduced that has everything you need to make this product class very useful, and you have a dominant design. In the case of typewriters, this was exemplified by the Underwood Model 102. But when you get to a dominant design, innovation stops. All viable entries into the market, must now have these standard features. In fact, it's hard for you to get any additional innovation in there. The "WIMP" [windows-icon-menu-pointing device] interface, for example, hasn't changed fundamentally for 30 years. I thought I'd bring a couple examples just to illustrate this. I won't go on very long about this, but I thought it might be illustrative. This portable typewriter is the "laptop computer" that I had in college and that my mother had in nursing school in the 1930's. It reflects the main features of the dominant design, even though as a portable, it was an extreme design point.

Yost: Yes, I actually grew up when taking typewriting in high school was standard and typed on a manual.

Card: One of those Underwood things.

Yost: It wasn't Underwood. I think it was Smith-Corona, but it was a manual typewriter.

Card: So you had experience using one of these?

Yost: A manual typewriter, yes. With a ribbon.

Card: The ones that come after it, a little bit easier.

Yost: Yes. That definitely is older than the one I used.

Card: It might be argued that two-color ribbons—black and red were part of the dominant design. Here's the keyboard from an Alto. And I want you to type on these keys. Isn't it great you have that much key- travel and the keys are sculpted to give you tactile input on where your finger is relative to the key? Keys like this that optimize typing performance used to be part of the dominant design, as opposed to some current designs that emphasis the thinness of the computer.

Yost: Yes.

Card: This is coming to a point soon. [After showing the Underwood typewriter and some computing hardware artifacts]. Now here, this keyboard is a little bit eccentric. That one there is less eccentric. And this one—this is interesting because this is a Mac keyboard, but it's the IBM PC keyboard layout. And you can see how in a company, they might want all keyboards to be the same and so they just buy these like this. This is an example of the invisible hand of the dominant design, because there's no functional reason to have this keyboard on a Mac. A whole bunch of these keys have never been used on a Mac--EVER!! They've never had a meaning, and yet, they're required. The SysReq key, for example, derives from the IBM 3270 mainframe console where it was used communicate with the operating system directly and it was used on the IBM PC/AT personal computer to switch among operating systems PC/DOS, PCM-86, UCSD p-System, and Xenix. If you are a Mac user, I bet you never felt a strong need for this functionality. The SysReq on the Mac keyboard never did anything! It is there because it got swept up as part of the dominant design. And here is the keypad, and 7, 8, 9 is at the top, isn't it?

Yost: Yes.

Card: I got to design the Xerox Star 2 keyboard. Not because I knew anything about keyboards, but because there wasn't anybody else to do it. I've always thought that numbers 1, 2, 3 should be at the top, like they are on a touch telephone. But on an adding machine, they're this way, 7, 8, 9 at the top. Not many people touch-type adding machines (except for the small fraction of the population who are accountants). So, I was determined I was going to fix this little oversight of the world, that the keyboard should have 1, 2, 3 at the top, and that's even supported by empirical data. But, as I sat thinking about it, if I designed a keyboard with 7, 8, 9 at the top, nobody would ever notice. Xerox would never be criticized for it. But if I put 1, 2, 3, at the top, even though it's a certifiable improvement, some people might say, "Oh, that does not conform to standard," and there's a possibility that there'd be a negative reaction in the market place, since it's a change. So even though I wanted to, and had the power to change this, as I sat thinking about it, why would I ever put a very expensive, major corporate product like the Xerox Star at risk—

Yost: Right.

Card: —for a detail like this?

Yost: And people who had worked in retail, at grocery stores, they were used to this layout.

Card: Yes. And suppose there's some big company, Walgreen's or something, that wants their Mac keyboards the same as their PC keyboards.

Yost: Right.

Card: Then there's some hullabaloo that has to take place in order to do this. So, when I realized that, I realized that the invisible hand of the dominant design was keeping me from changing this because once it's standard, stuff just stays in place. For a while. Now, you can change this if you have a big enough improvement, but the Windows-Icon-Menu-Pointing device interface has gone largely unchanged for 30 or 40 years even though there's major mistakes in it.

Yost: Right. I mean, there are less common alternatives to the QWERTY keyboard—

Card: Yes. Well it turns out because I have a way of calculating that, too. I can predict that the Dvorak keyboard, the alternative arrangement everyone usually brings up first, only gets you about 5% improvement. And the reason is that the likelihood of the keys jamming, when the QWERTY keyboard was first defined, was greater when keys next to each other were struck in sequence. That lead the placement of the keys so that the pairs of key hammers that are next to each other are struck with low frequency and high frequency key letter digraphs have hammers not near each other—like on different hands. So, in typing the frequent word T, H, E. There's 'T' on the left hand and 'H' on the right hand and 'E' on the left hand—the keystrokes are all between-hand. Almost all the speed in typewriter keyboard design is in how you do the between-hand key pairs. In between-hand keystroke pairs, you hit a key and the next key is going to be hit with a finger on the other hand. That allows this finger to be going up, while the other hand's finger is going down, and they can overlap each other in time. So, once you realize that most of those opportunities are already taken in a QWERTY keyboard as in the T, H, E example, if I go and try to beat [inventor Christopher Latham] Sholes [QWERTY] keyboard with a new design, there just isn't that much upside left to get. And what's more, the new design will only work for people who use the new keyboard exclusively. The [inventor August] Dvorak keyboard typists aren't allowed to type on any other keyboard arrangement, because otherwise, there's interference in learning. So that's another example of theory influencing the design of products. To come back to the point I was trying to make a long time ago in our conversation, a major reason why things don't seem to change much at the CHI conference some years, is that the phenomenon of the dominant design is holding them back not that CHI doesn't have lots of opportunities to do new stuff. This needs to be taken account of when deciding where effort is likely to have the best payoff.

Yost: Do you recall if there was any on-going debate about the name SIGCHI, putting the 'C' before the 'H', computer before human, in the name or was that just to create a pronounceable?

Card: No, I don't think there was much debate about it. I mean some people were annoyed by having "computer" first, but it's irresistible for the computer scientists. I thought long and hard for our book about which went first and which words to use. It was obvious to me that putting the computer first in a naming a field whose whole rationale was to put the human first was self-contradictory. But the opposite choice might be more salient within computer science, which has a genetic weakness for clever abbreviations. The same issue comes up in persuading women that we want their serious participation in a field we have just named “man-machine communication”. It’s not fatal. We can try to explain our genetic weakness for a good alliteration. But life is easier--and the field easier to market--if you wear key values on your sleeve. Notice that even though the field is called CHI in computer science, we still use the term “HCI” a lot.

Yost: Right. In 1986, you moved from various grades of the research staff to become a principal scientist and then a couple years later, the manager of the user interface research group. Can you tell me about those changes for you into managerial roles, and what you looked for in a successful scientist on your team?

Card: Actually, Xerox had a dual ladder system and I eventually became a Senior Research Fellow, the highest grade on the technical ladder. At the same time, I was an Area Manager on the management ladder. These dual positions enabled me to control resources without being swamped by management. I definitely held up the banner for science. My group, User Interface Research (UIR) was slightly odd-ball, because I wanted people who could do cognitive science, visualization, systems-building, and business. You’ll get an idea of the mix of the group if I give a partial list some of my group and a gloss of a few things they did. Peter Pirolli (psychology, cognitive modeling) had been a graduate student in the CMU psychology department. He knew the CMU Newell and Simon world view and was probably the only one who really understood what I was trying to do. Mark Stefik (AI) is one of the smartest people I know. He defined a digital copywrite system while in my group that was adopted by the industry and sold by Xerox for hundreds of millions of dollars. Later, he became my boss. Dan Russell (AI, Learning) is famous for the NoteCards system, a tool for composing and using knowledge databases. He was one of the leaders of our sensemaking effort. One of my great opportunities came when George Robertson (systems, visualization) joined to us from Thinking Machines Corporation where he was writing the operating system for The Connection Machine, a parallel processing computer. George had been a

Newell graduate student at CMU with me. George could build anything. He and I collaborated on many systems: The Cognitive Co-Processor, the Information Visualizer, the Cone Tree, the Web Forager, a 3D copier interface, and many more. Jeff Heer (visualization, systems) spent a year with me as a postdoctoral student. Jeff developed a visualization called the Degree-of-Interest Tree in which node-sizes or other display properties change dynamically as the user's interest changes. He managed to fit a million virtual nodes on the display dynamically interacting at animation speeds. Then he packaged the system into a toolkit called prefuse and distributed it world-wide. Ed Chi (visualization, systems) and Jim Pitkow (psychology, systems, business) were such hot students, that I hired them while they were still in graduate school, thereby snatching them out from under the nose of Microsoft. They finished their PhD degrees at PARC with me as a member of their thesis committees. Because of them, we could do large-scale Internet computations before almost anyone else. Jim Pitkow loaded the entire Internet into a new machine with huge disk space that we bought for the purpose, and this allowed us to compute node statistics from which we characterized the structure of the Internet and together with a theoretical physicist, Bernardo Huberman, establish the Lognormal Law of Surfing. Jim and Ed quickly learned how to hook up racks of machines so one of their computations might run for a week with, say, 20 computers running in parallel. Ed wrote a visualization, the Disk Tree, that visualized simulated user searches using a spreading activation semantics representation of the entire website. This permitted us to run groups of simulated users doing test tasks and visualizing, for example, where they would get stuck on a website. Lechan Hong (visualization) collaborated with me to design and build my favorite visualization, a 3D animated book called the 3-Book. Bongwon Suh (visualization, systems) and Bryan Pendleton (visualization) collaborated with me to build a visualization of time-varying hierarchies. Eric Bier (visualization, systems) built an entire application for aiding intelligence analysis. Ramana Rao (visualization, business) built visualizations (the hyperbolic tree, Table Lens) and founded a startup (InXight) to commercialize our visualizations. Jim Pitkow founded a startup (Outride) to commercialize our search technology. Walt Johnson (AI, education, business) created electronic paper user interfaces, and started a new division of Xerox to commercialize them. Bill York (Systems, Business) served as a development manager. For example, our original target for our InXight startup was the Web Forager. Bill hired four developers, defined the specs, and managed the development. Once I had George, I added Jock Mackinlay (AI, visualization) to the group. Jock had written an AI program for his thesis that automated a semantic system for visualizing data. So, I essentially hired Jock to take that research and go forward on it. That gave us a way of going into the information visualization area with a theory.

Yost: Can you expand upon that research area?

Card: Yes. Jock's thesis was based on [Jacques] Bertin's visualization semiotics theory. Jock's formulation is that you have data on the one hand, and a visualization on the other, and the visualization has to follow two principles. One of these, the *Principle of Expressiveness*, is that all the relationships in the data have to be present in the visualization and there can't be relationships in the visualization that aren't in the data. It's easy to get what you might call semantic aliasing--implied relationships that sneak into the visualization but aren't really there in the data. Let me give an example. Suppose I have some data that shows the number of people who purchase different models of cars--Fords, Minis, etc.--and I plot the data on a graph as a bar chart. The order of cars might be arbitrary. Any patterns that I see in that arbitrary ordering is meaningless and the visualization is said to be semantically aliased. That visual pattern not meaningful. It's only because of the requirements of space that the bars in the bar chart show up in some order. Now, if I can give some order to the cars, like purchase price, then I can make the visual relations correspond to the data relations. How do I represent things in meaningful visual fashion? Well, I have two or three dimensions to work with. And then I have things like sizes of dots and color and so on. These are called retinal variables. The second principle for data visualization, The *Principle of Effectiveness*, is that the visualizations must be easily interpretable by the human perceptual system. Some techniques for representing the same data are more efficient than others. I can tell if the data forms a sine curve much more easily if I use position on the display to represent the values in the curve than if I use color. I can also do various distortions of the visualization. Here are some concrete examples [showing a set of images]. Here's a visualization called the "perspective-wall". The wall is like a long ribbon that's divided into three parts, the center panel for detail and two perspective panels for context. The ribbon animates horizontally, bringing part of the data into detail focus, with the other parts serving as context. This lets us explore data with one long dimension, like hospital patient histories or timelines that are three times larger than we could normally get on the display. Distorting the data in the side ribbons of the display allows us to fit in more data, but having these distortions be according to a perspective transformation means that the user doesn't even notice the transformation is a distortion. We have a whole set of exotic visualizations like these. One of my favorites is the hyperbolic tree built by Ramana Rao. Here's an example of the hyperbolic tree [shows Hyperbolic Tree]. Here's my point of attention and now the things get smaller as they recede from that point of attention. The idea is to see if we can squeeze more information on the display by using distortion and motion. With the perspective wall [shows image, we can get three times as much information on a single visualization, but on the hyperbolic tree we can get maybe a million times as much information.

Yost: What about you and your group's collaborative work on information foraging theory. Can you talk about that?

Card: Yes. That's an interesting one because at PARC, my manager, Per-Kristian Halvorsen and I got together and sort of decided that PARC should concentrate its research more on digital documents and information retrieval, because PARC had already completed its original research missions and it was time to renew its vision. PARC starts out with three missions. Did you know this?

Yost: I think so, yes, but for the recording…

Card: Ok. One of them was expansion into office automation. The problem was that Xerox was quite large and had too narrow a product base to support that large a company. So, the idea was to branch out into office automation and the office of the future. The second task of PARC was that Xerox was an analog company, and the future was digital, so PARC should lead that digital transition. Take a copier for example. How would a digital copier be architected and what new features could it have? And the third mission of PARC was that Xerox was based on Xerography, but there were other marking technologies that were possible, like acoustic inkjet. And if one of those came up and hit, Xerox should be a leader in that technology. And those were the three missions. They are fairly simple to say. And by 1980, PARC had done all of them successfully. What did you ask again? I'm sorry.

Yost: I'd asked on the research on information foraging, the area and its impact.

Card: Ok. So, Kris Halverson, my boss, and I, led this committee on future PARC research directions and came up with recommendations for doing research in information retrieval. We wrote a report that laid out a research plan. There were some resources attached to that, I got one additional person and Kris got two. I should say one background thing first. Xerox, by now, had fallen out of the office automation business and the workstation business, something that bummed us all out. PARC was pretty good about delivering something that's concrete to do. So, we decided that, well, we tried to get some coherent future planning information out of Xerox, but we finally decided on a strategy which I called "lead the duck". If the duck can't tell you where it's going, shoot in front of the duck, and hope the duck runs into it. Our logical conclusion for where Xerox would have to go was into digital documents. Otherwise it was going to be like Kodak which, even though it was making a lot of money with film, was headed for disaster. So, we figured that if Xerox went into digital document products, there would be things that it could do and had developed skill sets for and that this would hook up nicely with its digital printers and copiers businesses and with its new initiative to be "The Document Company". Our take on information retrieval was that previously people had gone after how to improve recall and recognition. *Recall* is how much of the good stuff out there did you find and *recognition is*, how much of the stuff you found is good stuff? So, we have these zippy computers now and according to these metrics, if you made your information retrieval system go a thousand faster, you would've made no improvement because you didn't affect recall or precision. An interesting alternative way of framing the goal was to invent systems the maximized the rate of information gain. Lead by group members Mark Stefik and Dan Russell, we looked at a task where you collect information about the technical specs of Xerox printers because we wanted to make a training course for Xerox repairmen or salesmen. As you go through the logical steps, you lay them out in some diagram, or you formulate some schema and then you apply that schema to all of your documents and note some "residue" that doesn't fit within the schema. Then you fix up your schema so that it doesn't have those errors anymore, a process called schema repair. Then you reiterate and do it again. This time there is less residue, and you keep iterating until the residue is down to an acceptable level. We call this the "sense-making process", and it turned out to apply to lots of things. We used it, for example, in our studies of intelligence analysts because they are information retrievers par excellence. We also used it as the theoretical backbone of a whole new area of research funded by the government called Visual Analytics, which now has its own scientific conference. We came up with a theory of intelligence analysis that Peter Pirolli and I published in a professional intelligence journal. Sensemaking went beyond recall and precision by defining things in terms of the rate of constructive steps.

Yost: Right.

Card: . This emphasis on the cost or speed of finding information also sent us off on another direction—Information Foraging Theory. Peter Pirolli suggested looking at optimal foraging theory from the field of environmental biology. This theory describes how patterns of feeding and other animal behaviors can be predicted. After some experiments, we found that we could use analogs of the optimal foraging theory methods to describe information use. Maximizing the rate of information gain for information seekers was a lot like maximizing the rate of energy or food gain for animals. For example, one often-quoted result from optimal foraging theory, Chernov's Marginal Value Theorem, deals with situations in which foraging within a patch of food has a decelerating cumulative gain function—as time goes on, it gets harder to find the choice pieces of grass. The theorem predicts that a forager is a food optimizer and should remain in a food patch until the marginal rate of found food drops to a rate that is less than the average rate for the environment, including travel to the next food patch. The moment this happens, the forager should switch to a new patch. Technically, this is calculated by plotting a curve representing the gain of food over time, including the time needed to travel to the food patch, then drawing a straight line from time 0 to the point tangent to the food gain curve. The point of tangency is the point at which the animal is better off switching patches.

Information foraging is similar to food foraging and inspired us to work out how this similarity this could bring insight to information access. For example, we can improve information foraging by within-patch enrichment, like filtering the information with a computer, or we might make improvements by between-patch enrichment, like rearranging the office so that the information sources are closer together. Now we can apply the Chernov Marginal Value Theorem to show that within-patch enrichment will reduce the average time a user will spend before switching to an alternative source. Now we are in a position to back-project solutions from optimal foraging biological domains to invent improved information appliances. We can take the fact that these theories give us abstractions for talking about the problem we are trying to solve, the phenomena of nature that we can use to solve it, and the technology that embeds this phenomenon—the abstractions we need in order to grow a field of study and design. We can make progress without starting from the beginning each time, or merely resorting to appropriate-and-improve previous designs.

Yost: And you taught at Stanford, you were an adjunct in the 1980s. Did you enjoy teaching students there?

Card: Yes. A lot. I was also recently a visiting professor in the Stanford computer science department and then an adjunct professor n computer science.

Yost: And can you discuss your style and your role in mentoring students?

Card: Yes. I guess the thing I'd done recently is I had a number of PhD students, who were writing their theses. It's a larger project than they've been responsible for before. I used to have talks with them about what their difficulties were in order to help them, and it turned out that they tended to have the same set of difficulties. About the same time, because I was in the HCI group at Stanford, the IEEE Visualization Conference asked me to chair a committee that chose the best visualization thesis for the year. So, I ended up having to read 20 theses, which is a lot, and our committee chose the best. Then, since I had read so many theses, they asked me to give a dinner talk at the conference on what makes a good thesis. So, I wrote an article that goes through how to win at writing a thesis, and that seemed to help my Stanford students. They all got out. Although I've been on a number of thesis committees, I'm sort of out of the main line of fire, so sometimes they can do things with me that have fewer consequences.

Yost: You've won so many impressive awards; you're an ACM Fellow, with a lifetime achievement award at SIGCHI, both in 2000, The China Knowledge Center for Engineering, Sciences and Technology, ranked you as the number two most influential scholar in Human Computer Interaction. Can you tell me what these awards have meant to you, and what these organizations have meant to you?

Card: Yes. Well, I have been fortunate. I have a rule: when you win one of these things, you are allowed to spend one day celebrating and really enjoying it. You really should take some time out, and, say, go to a baseball game.

Yost: Yes.

Card: Then the next day, it's time to get your ass back to work. Don't pay any attention to what you don't win. You can always go to a baseball game anyway. Remember that there is more randomness in life than people appreciate and there is research showing the people tend to take credit for the randomness that goes in their favor, but to blame circumstance for the randomness that doesn't.

Yost: What do you see as the greatest opportunities and the greatest challenges for HCI moving forward?

Card: Well, HCI is becoming very much design oriented, where "design thinking" is inspirational. That's good in some sense. HCI has trained all these Masters students to go out in the industry and do good stuff, but I really think it needs to focus more on the theory piece, otherwise it will have sandy foundations like those methods in the Chapanis non-experimental methods study we discussed at the beginning of this interview. That's what I've been trying to nurse along, and I'm still going to try to do something about that. I think most of the people in HCI don't think about this or necessarily agree with me on the theory statements. But that just sets my job back to where we started this brash process, that just eggs me on to make it happen. Remember, I had two goals here. One was to try and develop a psychology that you can use to make practical calculations with, that you can use in engineering. HCI is just the example domain for testing this idea. But as time went by, I did get captured by the focal domain. So my second goal was to develop a way to bring some theory and method to the design computer interfaces--to design interactions between machines and humans.

Yost: Before we conclude, are there any topics I haven't brought up that you would like to discuss?

Card: I'm working on an interesting area now which started out as a medical thing. There's a lot of medical errors that are being committed and why is that? Well, it turns out that medicine is so intrinsically complicated. There are so many things to know and do with great precision. When you try to solve that by specialization, you lose the big picture. So we tried to externalize the cognition. We designed some checklists for emergency situations, because checklists have been found effective in reducing errors, starting in aviation. But many doctors dislike checklists because they are too slow. So, we tried to make ours really fast. And in the process of doing this, we took eye-tracking data from about a hundred MDs. We looked at how the doctors move their eyes when looking at these checklists. They move their eyes so that they look at this visual patch and then this patch and this patch, looking for the information they are seeking. Well, that looks like it's describable with information foraging theory. And there's also somebody at Harvard who's now used the information foraging theory for visual search. That would be confirming of our earlier work. We built a device that makes images out of these things, and eye movements are really neat cognitively because there's a theory which may or may not be right (but I think it's probably approximately right) that you do all the cognitive processing on one eye movement before you go to the next eye movement. You don't have lagged cognitive processing. Well, this theory then gives us a language for several things that make the searches easier so we can play the game that we were playing before on the Rooms windows application, but now with information foraging theory.

Yost: That's fascinating to hear, as is all you've discussed, and all you've done, all the exciting research. Congratulations on all these amazing accomplishments! It's been a great honor to interview you!

Card: Oh, I think it's a great honor to have you come out here.

Yost: Well, thank you again.