Oral History of Edward Feigenbaum

Interviewed by:
Don Knuth

Recorded: April 4, and May 2, 2007
Mountain View, California

CHM Reference number: X3897.2007

Don Knuth: Hello, everyone. My name is Don Knuth. Today [April 4, 2007] I have the great privilege of interviewing Ed Feigenbaum for oral history where we hope to reach people generations from now and also people of today. I'm going to try to ask a zillion questions about things that I wish I could have asked people who unfortunately are dead now so that students and general scientific people, people with general interest in the world, historians and so on in the future will have some idea as to what Ed was really like even though none of us are going to be around forever. Now this is the second part of a two-phase study. Ed grilled me a couple weeks ago and now it's my turn to do him. I hope I can do half as well as he did for me. I'd like to get right into it But first of all sort of for organization, my plan -- who knows if it'll carry it out -- is to do a little bit first that's chronological, in order to set the scenes for his whole life. Then most of my questions are probably going to be cross-cutting that go through the years. But first of all, Ed, do you have some remarks you would like to start with? Or shall I get right into the—

Edward Feigenbaum: No, Don. Why don't you just get right into it. I have lots to say.

Knuth: Right. I'm always very interested in the origins of things -- the seeds -- and I spend more time reading chapter one of a book than chapter ten always. So you were born, and I'd like to launch this whole thing by saying, even before you went to college, what was it like growing up? I understand you were born in New Jersey.

Feigenbaum: I was born in Weehawken, New Jersey, which is a town on the Palisades opposite New York. In fact, it's the place where the Lincoln Tunnel dives under the water and comes up in New York. Then my parents moved up the Palisades four miles to a town called North Bergen, and there I lived until I was 16 and went off to Carnegie Tech. To try to paint the picture for you of those early times, let me paint it culturally first and then let me paint it as I saw it as a scientific kid. Culturally, it was from a Jewish family in the Jewish culture of New York that happened to be living in New Jersey. But they had come from New York and Brooklyn -- the same Jewish culture that you know a lot about through many other people who have come up into science and other professions. That means there was a great deal of emphasis… That Jewish culture thinks of itself as the people of the book, and so there's a tremendous focus on learning, and the book, and reading, and all that. So I think from what I hear -- it's hard to remember back when you were a tiny kid -- but I hear that I got to learn to read very early. That's kind of the story of my life. It was a middle-class to lower-middle-class family, culture. Nobody was very rich. We certainly weren't in my family, and it was all a public school education.

Knuth: Did you have a large family?

Feigenbaum: No. It was myself, born in 1936, and then my brother -- I don't know what you'd call him, my half brother or something -- in the end of 1946. My biological father actually died when I was an infant. He died of an accident or something, not quite sure in the family what he died of, but he was gone by '37. My mother remarried, and then after World War II that's when my brother came, because they were holding off not knowing what was going to happen during World War II. Although it doesn't relate to my professional career, I think there was really quite an impact of being a Depression baby and quite an impact of being a kid who grew up through World War II. It wasn't like, let's say, my wife Penny, who actually lived in Tokyo during the air raids. It wasn't anything like that. But it was still a major impact on your life to be living through this gigantic war effort.

What sticks in mind is my stepfather, whose name was Fred Rachman, was really the only one around in the family who had any college education. CCNY. He used to take me off to the American Museum of Natural History, which was right across the river. Go into New York on the bus, and go up on the subway. Every month the Hayden Planetarium would change its program. They don't do it anymore because of cost reasons; they change maybe every three months or six months. But every month we'd go over and see the new program. We would examine one more room of the American Museum of Natural History. A tremendous impact on my life, especially the astronomy part, which is not unusual for kids, getting into science via astronomy. The other thing I remember from those early days -- I'm talking about 10 years old or less -- is being a young science nerd right then and there. It's amazing how fast that develops. That means reading books like… Well, many times you'll hear our generation of scientists talk about being motivated to go into science because they read "Microbe Hunters". Well, I did read that, but in my case it was making sure I read Scientific American every month as a kid.

Knuth: I notice that you were born two years earlier than I was but you entered college four years earlier than I did. So you must have picked up two years on me somehow. How did this happen?

Feigenbaum: Acceleration through school -- the teachers wanting to put me up a class to keep my mind busy, and my parents going along with that. I know that my daughter doesn't allow that with her children. She wants her children maturing at the same rate as everyone else matures. But my parents were okay with that, so I graduated high school when I was 16.

Knuth: You skipped fifth grade or something?

Feigenbaum: Somewhere in there. I don't remember which ones. There were two of them.

Knuth: Being born in January tends to set people back, so maybe they could pretend that you were born in December.

Feigenbaum: Yeah. Maybe.

Knuth: Studying science is right there at your fingertips in New York. I had to go to Chicago. I could go once every three years to see a planetarium. But what about languages? Did you also study-- I noticed that when -- we'll get into it later -- when you were in Russia you spoke German to some of the scientists. Yu must have studied some languages also.

Feigenbaum: Well, languages: I'm not very good at languages. I tend to shy away from the things I'm not very good at including,for example, the tragedy that Penny and I… Penny's Japanese, and since we were married in the mid 1970s and I travel a lot to Japan, I should have learned Japanese back then. It would have been very fruitful in my life, and would have made my life much better, but I never did. So where did German come in? Well, the family spoke Yiddish. Yiddish is kind of a variant of German, so I had some of the words in my head. I had to do something in high school. You have to take a language, so I just did German because it was easier than doing French or something else [like] Latin. By that time they weren't teaching Latin I think in my high school. So I had some German stuck away in my head from high school German. Then came these trips to the Soviet Union in 1960 and 1964. I really attempted to learn some Russian before I went there. I actually hired a graduate student at UCLA. I was working at Rand Corporation in the summer, and I hired a UCLA graduate student to teach me Russian. But there is an interesting lesson there for AI people -- cognitive scientists and all of that -- which is the student taught me the rules of grammar, of Russian grammar. Because that's what he knew, and that's what his expertise was. Therefore when I got to Russia I could say nothing perfectly, which is the stupidity of just using the rules. So I'm not real good at languages.

Knuth: That may be the same reason I'm not good at languages.

Feigenbaum: Let me also tell you an interesting little anecdote about languages. In graduate school in the Ph.D. program we still had a requirement for a language at Carnegie Tech in those days. I was kind of pushing for: "Why isn't my language Fortran, or IT, or something like that? That's my language." No, it had to be a real human language. So I did German, and what I took as my text was Freud. Well, Freud is really hard, and I didn't pass that German exam. So next time around I said, "Okay. I'm just going to do a statistics book." Of course math is the same in every language so it was trivial to translate the statistics book. I passed my German exam.

Knuth: Only one language at Carnegie?

Feigenbaum: Yeah, one language. Oh, I wanted to add one more thing about growing up. There was one key thing that I didn't mention before. Absolutely key. When I mentioned that my stepfather, Fred, was a CCNY graduate, he was actually in accounting. After his degree he went to work as an accountant in a firm, and had and owned a Monroe calculator. Initially the Monroe calculator with the little flip thing where you flip the digits from position to position, but later on a motorized one, which was at that point not a Monroe but a Frieden, I think. I was absolutely fascinated by these calculators and learned to use them with great facility. That was one of my great skills in contrast to other friends of mine whose great skills were being on the tennis team. They would get varsity letters on the tennis team. They would be on the basketball team. Not me. Me, it was the calculator. I would actually give demonstrations of this calculator by dragging this really heavy object -- if anyone in the video audience wants to see one, come to the Computer History Museum and you'll see some of these 30-pound things -- dragging it on the bus down to school, so I could take it to school and show people how to do this kind of stuff. So I was kind of primed to like calculators.

Knuth: You mentioned "Microbe Hunters", and astronomy, and calculators. But when you finally got to Carnegie you majored in electrical engineering. Did you also have some radio or something?

Feigenbaum: Yes. That's right, all the normal kit stuff. My parents didn't want me to listen to the radio when I should be going to sleep, and there was the Knicks game from the Madison Square Garden, so I made myself a little crystal radio with earphones. They thought it was a great science kit that I was doing. Actually I was just putting together this crystal radio so I could listen to the Knicks game. Why did I go into electrical engineering? That's partly because in this middle-class to lower-middle-class culture people were focused on getting a job that would make money for the long run. No one ever told me there was a job, for example, called physicist. No one ever told me there was a job called symphony conductor.

Never. No one ever mentioned that that was something you could do. Engineering was something that, if you're good at math and science, you can make a living at it.

Knuth: You did have some kind of a role model? You knew somebody who was an electrical engineer?

Feigenbaum: No. Just talking around.

Knuth: Vocational exams? They would say, "This looks like a good career for your skill set"?

Feigenbaum: Yeah. I think there is the other side of that coin too. In talking about a vocational counselor or even a college prep counselor, because I was going to a school… North Bergen didn't have a high school, so I had options to go to different schools. The one I chose was the one that was known for having a college prep orientation. They had a college prep adviser. I am amazed - here I am going through high school with straight A's in math, science, all the geeky stuff. She never once mentioned the initials MIT. Never once, not a single time. Never mentioned the word Harvard. Nothing. Carnegie Tech came up because they were advertising these Westinghouse scholarships and wow, that was something! My family didn't have much money so that was a way. Why don't I apply for one? I didn't actually get a Westinghouse scholarship. I got the equivalent of it, which was a Carnegie Tech scholarship, incidentally, at that time $680 a year. Compare that to Stanford's tuition!

Knuth: That was a lot of money.

Feigenbaum: Yeah, really a lot of money.

Knuth: I don't know if you remember when you took these aptitude tests. Did it also tell you what would be your worst career? I remember for me it was veterinarian.

Feigenbaum: I just really don't remember it at all.

Knuth: Carnegie must have been very competitive. Certainly, I don't think this adviser would have advised many students to go to Carnegie at that time.

Feigenbaum: Right, but it turns out life is strange. Life is an interesting set of choices. The decision to go to Carnegie Tech was a fantastically good decision. If it weren't for that, I wouldn't have met Herb Simon, I wouldn't have met Allen Newell, I wouldn't have met Alan Perlis. But it was just a pure piece of luck at the time, just because they were doing PR on the scholarship and no one mentioned MIT to me. I might have gone to MIT, and I might have met Minsky and I might have met McCarthy.

Knuth: Or whatever. You might have gone into biology or something, but certainly you had a perfect match where you went. I'm getting a little ahead of the story but I think actually Simon came to the business school rather than the… As an undergraduate you're studying power systems and various things that electrical engineers studied at that time, not computers. Is this correct? Now you go through four years of traditional studies, and I suppose electronics.

Feigenbaum: Interesting time. It was the time of a switchover, at least at Carnegie. Maybe MIT and Stanford may have been a little bit ahead of them, but at Carnegie it was the time of the switchover from power engineering to electronics. The only electronics offered was a senior sequence in electronics. That's it. Before that, it was all power engineering and thermodynamics. But thermodynamics wasn't physics thermodynamics. It was steam tables.

Knuth: Steam tables!

Feigenbaum: Yeah. Pressure boilers and things like that. At the end of my freshman year I was doing really well, but I got to feeling like there's got to be more to a university than this. So I got the permission of my undergraduate adviser, who was a really broad-minded guy, a wonderful guy. He ended up, I think, being provost at Carnegie Tech. Ed Schatz. I had looked around in the catalog for what's interesting to take in this university. Well, this university was pretty broad. It had an art school. It had a great drama department, which incidentally led to ACT in San Francisco. Bill Ball was a student while I was, and then he later came out and started ACT. It had a great drama department, a great painting and design department, architecture and so on. There were a lot of things to look at. I looked through the catalog and found a course called Ideas in Social Change. I don't know why that hit my fancy, but I asked my adviser could I take that and he said, "Sure." He said, "With your grades you can take anything you want."

It turned out to be in a place called the Graduate School of Industrial Administration, sort of a relatively brand new building. I guess it had been started by Mellon -- Mellon Foundation or something. They had gotten one really top-notch economist, Lee Bach, to be the founding dean, and Lee Bach had chosen a couple other people to help him bootstrap this school from nothing into something. One of those people was his friend, Herbert Simon, who was a behavioral scientist who had published Administrative Behavior at that point and was also doing mathematical work in behavior science models. The other one was Cooper -- remember Charnes and Cooper Linear Programming Applications? They bootstrapped the school into existence. It was a miraculous decade or so. They started the school in the late 1940s, early 1950s, and between that time and 1965 or so they had already produced the material for four Nobel Prize winners including Herb Simon, Franco Modigliani, and a couple others. Dick Cyert became president of Carnegie Tech, [now] Carnegie-Mellon. An amazing place. It was partly a school in mathematical economics, partly a school in mathematical behavioral science, mathematical models in the behavioral science, and partly an MBA program to earn money for them. As an MBA program they were the first ones to introduce quantitative methods, namely to take a stance against the Harvard case study method of learning what an industrial administrator would need to know.

Knuth: Instead of case studies they were talking about…

Feigenbaum: They were talking about simplex method. They were talking about ideas in social change, which was taught by an instructor who had just come from Yale, Jim March, who is now a retired famous Stanford professor. March started out that course by giving us Von Neumann and Morgenstern to read, the game theory book. Wow. For a sophomore this is really mind blowing.

Knuth: Hard science, or something quantitative rather than qualitative.

Feigenbaum: Yeah.

Knuth: As an undergraduate you're taking this as an elective. Simon is a regular faculty member, or a visitor?

Feigenbaum: No, Simon was one of the founding members of the school. He was of course a full professor, but he was also associate dean of the school. At Stanford we might call him associate dean for research. He wasn't really a very good dean. We know that. Experimentally, Lee Bach took a sabbatical one year and left Herb to run the school. Within, I think, a third of the way through the year it was clear Herb couldn't run that school, so they gave it to somebody else to run.

Knuth: As an undergraduate did you have any other thing besides meeting Herb that shaped your future life? For example, somehow you would have the ability to be a good dean yourself. Were you a leader of student organizations, or anything that gave you some experience?

Feigenbaum: Yeah. I actually did all of that. All of that extracurricular stuff was part of my life. Partly because it was interesting, and partly because, like I said before, I was good at that sort of thing and I wasn't very good at the football team. So I got to be head of -- the name of the magazine was called "Carnegie Technical", but it was like a Scientific American for Carnegie stuff.

Knuth: That's interesting.

Feigenbaum: I think you did that too.

Knuth: Well, Case just started in my senior year. So I had one year experience doing that, but Carnegie was ahead of us. There was the Sigma Xi Honorary… no, I'm sorry. I can't think of the name of it but for journalism there was an organization.

Feigenbaum: Yeah, I did that sort—

Knuth: **--**that sort of thing—

Feigenbaum: I also did student body president, I think. No particular reason. That was the last election I ever ran for and the last election I ever won.

Knuth: I ran for student government, but based on the fact that I had published an article in Mad Magazine. But I lost miserably. You even won elections. That's amazing. Had you done writing also in high school?

Feigenbaum: Nothing sticks in mind. Maybe. It's possible that I wrote something for the student… Maybe I was on the student newspaper. It just doesn't stick in mind. Writing became a skill that I decided early on I really had to do well. Now don't ask me why I thought that. Maybe it's because I was so engrossed in reading. But in fact it's a hindrance actually to be a perfectionist on the writing end of things, because then you face a piece of blank paper a little bit scared. You never know if it's going to come out right. Herb Simon used to try to convince me that hey, wait a second, this is like debugging a program. You write a lousy draft, and then the next one is better, and the next one is better, and the next one is better. But it doesn't feel like that when I sit down to write something.

Knuth: Engineers are usually faulted on their writing skills. Apparently, Carnegie took that to heart and at least had enough in place that you could come out of your undergraduate years there as a good writer.

Feigenbaum: Yeah. Then one of these other broadening things I tried to do when I was an undergraduate, I actually took a course over in the art school from a novelist. I'm trying to remember her name right now but-- Oh, God. I-- I'm blocking on it. She wrote many novels of which I think her most famous one was called David the King. She was quite good as a novelist and I thought oh, this is a lot more interesting than engineering, I ought to be a novelist. I started to write some things in this class. I think it was two semesters that I took that. She convinced me toward the end of the term that I really should continue on being an engineer, so I had a day job. If I wanted to be a writer I could write as a night job, and I think she was right.

Knuth: The teachers must have been right. You graduated in 1956 and at this point there are no computers yet at Carnegie Institute of Technology. Is this correct?

Feigenbaum: Yeah, that's right.

Knuth: It's also true at Case Institute of Technology which I entered in 1956.

Feigenbaum: No computers. The first computers showed up in the summer of '56. It showed up along with Alan Perlis and Joe Smith and Hal van Zoren in a tiny little office in the basement of the Graduate School of Industrial Administration. There is actually a story about that. I told it again at the Carnegie-Mellon 50th anniversary celebration of that event, but it's also written up by Lee Bach in that special issue of the ACM that honors the IBM 650. The math department didn't want to have anything to do with it. The physics department didn't want to have anything to do with it. It was Herb Simon who loved it, and Herb Simon found Al Perlis. Lee Bach and Herb Simon gave it space in the basement of the Graduate School of Industrial Administration, and that's how it got launched.

Knuth: That would be a big difference from MIT, I guess, where several computers had been around for…

Feigenbaum: Yeah. That's right.

Knuth: …for ages. When you did get a computer, when did you first see it? Were you there in the summer?

Feigenbaum: No. Well, I would have to go back a little bit to tell you sort of a born again story, which I will. But to answer your question directly I was working at IBM in the summer of 1956 in the 590 Madison Avenue world headquarters location except around the corner in a brownstone where there were several things going on. On the first floor there was the graduate student… Even though I wasn't a graduate student, I was on my way to becoming a graduate student, and IBM was nice enough to consider me a graduate student. But there were other graduate students in that first summer program that IBM ever offered, where they wanted to hire graduate students for the summer to train them up. First they sent me to Watson Laboratory at Columbia University to learn the 650, because by that time it was sort of public knowledge that Carnegie Tech would be getting a 650. As long as it was going to be getting a 650, I ought to learn how to program it. Then I took a couple weeks, I think, and then they sent me back to 590 Madison Avenue where they were giving 704 classes. Incidentally, at the IBM Watson Laboratory an interesting thing that most people don't have the experience of is learning how to do wired programming. Not symbolic programming but actually…

Knuth: Plug boards.

Feigenbaum: Plug board programming, yes, with 16 registers. That was a phenomenally interesting thing for a kid, for a geeky kid. But anyway, 650 and 704, and then around the corner to the brownstone to do some real work for the summer. That's how I met [John] Backus. His group was on the fourth floor of the brownstone and they used to have seminars for us graduate students. Once a week somebody would come down from above and tell us things. So somebody came down. It turned out to be Backus -- I didn't know him at the time -- came down and said to us kids, "Hey. You don't have to write these things, "CLA," anymore for "clear and add" or OO5 for "add to the register." You can write algebraic formulas and our programs will translate the formulas. Get it? Fortran." Wow! This was big news to us kids. Bob Bemer came down -- people don't remember Bob Bemer, but he was a fantastic programmer. He came down from upstairs somewhere and showed us tricks, all kinds of numerical tricks you could do on computers, or coding tricks whereby you could save several registers in memory. That was amazing to us kids.

Knuth: These were—

Feigenbaum: So then I finished the summer and went back to Carnegie and that's where I-- The machine was there.

Knuth: Already, then, when you're seeing your first computers you're also seeing the world class programmers of those computers. So you get to start at the top basically…

Feigenbaum: Yeah.

Knuth: …instead of from the angles—

Feigenbaum: Let me go back and tell you the preliminary to that story, because it's the one that I tell again and again and again. It's in ten different papers of mine. Herb Simon actually quotes it in his autobiography. It also happens to be true, which is nice. Stories aren't always true.

Knuth: I'll have to remember to ask you that: "Is this true?"

Feingenbaum: Yeah. No…

Knuth: I understand.

Feigenbaum: I had gotten linked up with Herb after Jim March's course. My avocation, other than engineering, was to do behavioral science things, and in fact I did. One summer I worked for March and to it led to my first published paper. It was studies of small group problem-solving, where the small group was making a decision, and we were studying them behind one-way glass so you could see into the room and measure things about what they were doing and so on. One of the people I got to know that summer happened to be writing a book with Jim March in the room next door to where we were doing all these experiments. I was doing the experiments, and Jim and a guy named Herb Simon were writing a book -- this famous book called March and Simon "Organizations". So I got to know Herb. Not only got to know him, but got to realize that this is a totally extraordinary person. This is not your ordinary person you meet. This is a remarkable person and even a young kid like me got to know that. I got to feel that.

Herb was giving a course -- Carnegie Tech had semesters, and he was giving a two semester course called Mathematical Models in the Social Sciences. In fact he had published a book on that called Models of Man, if you've ever seen it. It's a collection of Herb's papers on various kinds of mathematical and difference equations models, statistical models, stochastic rats[ph?], things like that. So I was there to learn all that. Bush-Mosteller models you've probably heard of in learning models. So I can tell you the story of what happened with Herb, but I also want to note that if you read Herb's autobiography he tells the story of that fall in which the ideas of the first heuristic program are coming together. Newell moved to Pittsburgh in September -- I believe it was September of 1955 -- and he and Simon really got to work on…

Knuth: Let me try to unscramble the dates. You're a junior—

Feigenbaum: This is when I'm starting my senior year.

Knuth: You're starting your senior year, '55, '56, and Newell is arriving from—

Feigenbaum: From Rand Corporation.

Knuth: --from the West Coast, and I understand Newell had been studying the same kind of thing at Rand -- the way pilots make decisions…

Feigenbaum: That's correct.

Knuth: …and Newell had a mathematical background as well. He was trying to make models of these things. So he and Simon were a perfect team.

Feigenbaum: They met each other when Simon was consulting for Rand. Simon was a management science and operations research consultant -- occasional consultant at Rand -- and met Newell. Newell was doing this decision-making having to do with air traffic control. Not civilian aircraft control, but military radar stuff. Newell had gotten intrigued with the computer side of things, not the human side of things, because they were simulating these environments. Maybe it was the first great, big simulation. It was Rand System Development Division, later turned into a company called System Development Corporation, which developed all this. It got too big for Rand, and they spun it off as a company. Newell and Simon and a programmer at Rand -- now of course a legendary programmer, Cliff Shaw, now dead too -- were talking about how these machines which appeared to be calculators could do anything. Of course from Turing you knew that they could do anything, more or less. Given certain constraints they could do anything. So Newell and Shaw wrote a paper -- I think it was 1954 if I'm not mistaken or maybe it was published in '55 -- on how to build a chess-playing program. Simon wasn't involved with that particular paper. Oliver Selfridge from MIT dropped in out of the blue, and he gave a paper at Rand on recognizing X's from O's. Oh, God, that's amazing. A machine can do that? Of course, that was sort of the birth of optical character recognition, and like any good AI field it spins off and becomes its own field by the mid '60s.

Well, Newell once told me that was the strongest influence on his early career was that Selfridge talk. He and Simon had been talking about ways of not only building individual models, but building a kind of a system of how to approach these kind of problems. Simon meanwhile had published two key papers. These are absolutely seminal papers for AI, and AI people never read them because they're in another literature. They're both in Models of Man. One paper is called A Behavioral Theory of Rational Choice, in which he introduced satisficing as opposed to optimizing. The other is called Rational Choice and the Structure of the Environment, which is to say that a lot of what goes into human decision making has to do with the specifics of how the environment is configured, not the generalities. In a later publication he uses the metaphor of the ant on the beach. The ant is pursuing sugar on the beach. The ant is actually a very simple problem solving system, goal directed, looking for sugar. But the path of the ant seems to be very complicated, because the ant happens to be going around every little grain of sand that is like a boulder for the ant. So when you look at a complex pattern of behavior there are simple mechanisms that could be underneath and let's go for the simple mechanisms.

END OF TAPE 1

Feigenbaum: Simon writes about this period in his autobiography and he mentions a specific date, December 15, 1955, as the date when it all came together. What does it mean, to come together? It means that they actually put together a paper program, he and Newell, which quickly got implemented in the language called IPL-1, which was not a language that ran on any computer. It was the first list processing language, but it ran in their heads. It was sort of tested on a group of students and other faculty who sat in a big classroom, and when certain decisions came around they would raise their hand, go down this branch and—

Knuth: A perfect way to study behavioral models of human beings too.

Feigenbaum: Yeah. Then it was Christmas break. December 15 meant we were just starting or in the middle of Christmas break. We get back from Christmas break, and we had a few weeks left in the semester before the semester ended, so we were still going to class. Herb walks in to our seminar… Incidentally, the seminar was six people. I don't exactly remember who was in the seminar, but I was. It's not huge. He walks in and he says these famous words: "Over Christmas Allen Newell and I invented a thinking machine." Well, that just blew the minds of… A thinking machine! What he could possibly mean by that? We sort of knew what "thinking" was, but what does he mean by "a machine", and what does he mean by "thinking machine"? In answer to that Herb just gave us a 701 manual, or a copy, I don't know. Anyway, he had a 701 manual he gave us all, and he says, "That's what I mean by a machine." He sort of went into this general idea of what he meant by "thinking by machine". I don't remember much details about that lecture or that seminar discussion, but I remember taking that 701 manual home that night, having never seen a computer before, and I read the thing from cover to cover. I remember I went all through the night, and by the time I finished reading that manual I knew what it was I was going to do. It had nothing to do with getting a job as a EE like my parents wanted me to do. It was: stay with Simon and do more of this. I knew I needed money in the summer. Simon got me a fellowship for the fall, and that was no problem, but I needed money to live on. My parents didn't have that much, so that meant working in the summer. Working at IBM. That was where the IBM job came in.

Knuth: That was great. In those days you could apply for graduate school and get admitted a little more quickly now. If you make your decision in January or February to go into graduate school in the fall, Herb could pull strings—

Feigenbaum: Oh, this was one on one with Herb. I don't know what the process was, but in fact the one on one -- it's a different kind of graduate school experience than you would experience now. There were no courses. As you know from '68 when you got here – '68? Is that when you got here? Anyway, you know that I was pushing the idea that there shouldn't be any courses in our department. That just came from Herb. Whatever you want to learn in courses you can learn on your own. The PhD is a research training apprenticeship. So it was just one on one with Herb. I had to pass some exams. There was a marketing exam. Allen Newell was a graduate student with me, so Allen had to pass these same exams. He and I studied together for the marketing exam.

Knuth: I believe he got his PhD right away -- '56 or something like that.

Feigenbaum: No. I don't think so. Maybe '57 or '8 but I know he and I were studying from a marketing textbook together to pass this marketing exam, but he didn't get it right away.

Knuth: I looked it up on Dissertation Abstracts this morning, and it was dated '55 or '56 and it was called Information Processing Machines or something.

Feigenbaum: Well, I could be wrong.

Knuth: I might be wrong. Certainly he was ten years older than you, but he had dropped out of graduate school and mathematics. He had started studying pure math at Princeton, and then he decided that that wasn't for him. So this was his second career. I agree that he got his degree with Simon, and undoubtedly still hadn't decided his studying days were over so—

Feigenbaum: No. He went to Carnegie Tech specifically, A, to work with Simon but also what got him… The motivation to go away from the West Coast was partly that he could get his PhD with Herb.

Knuth: Right, but I think he's also-- even if he has a PhD he probably is still interacting with you. I might also have misread the date on his thesis but—

Feigenbaum: I think you misread it, because he and I were studying together for exams.

Knuth: Of course there's a language exam too, as you mentioned before. So there was something besides just one on one work there in order to meet the university regulations.

Feigenbaum: Don, let me just add something about this period which to me as an experience was quite significant. It's a little story about the first paper I ever delivered at a scientific conference, and it was actually a real paper also. I was one of the few people in town who could program a 650. There was Perlis, van Zoren, and Smith, and another graduate student named Fred Tonge who knew how to program the 650, and me. So for a few months actually, until the rest of the campus got aware of this machine, we had it pretty much to ourselves. I learned the IT language very rapidly as soon as I got back to Carnegie, so I could do a little bit of magic that other people couldn't do.

I knew I had to pass an economics exam sometime in this graduate school experience, so I figured I'd better learn some economics. I took a course from Dick Cyert. Dick Cyert later became president of Carnegie Tech, later Carnegie Mellon. Excellent economist. Dick was teaching a course in microeconomics. Well, microeconomics has to do with how actual companies make decisions about things. Usually it's treated very coarsely by economists who only know about writing out equations that you can differentiate and set to zero and find a maximum or a minimum. But here was Herb Simon with a paper called A Behavioral Theory of Rational Choice. Let's put that into economics. I'm really not a scholar at that point, I'm just a student, but Dick Cyert and Jim March were scholars, and I could do the programming. So for my term paper for Dick's course the three of us did a paper in which we did American Can Company and Continental Can Company in a duopoly, making decisions about the price and quantity of things they were going to put on the market. We were making predictions from the simulation written in IT as to what would happen in that market, and then testing that against what actually did happen during the American and Continental Can duopoly. It was really great. That kicked off a big project that later resulted in a book by March and Cyert called "Behavioral Theory of the Firm", and it led to many theses. Not mine. I was being "Simonized" into doing human information processing but March and Cyert continued it on in economic theory and when we wrote the paper they… Well, the first thing that happened is we sent it in to the American Economic Review, 1958, and no way they're going to… This was not economics. A computer? You couldn't use a computer to run a simulation to do economics. That was ’58. In ’59. They had already gotten the clue, and they ran a special issue on simulation, but we had to publish it in another journal called Behavioral Science. But that was a significant publication, and those two guys, March and Cyert, gave me the job of presenting the paper at the American Economic Review December conference. That’s where I got to do my first public presentation.

Knuth: You’re 21 or 22?

Feigenbaum: Well, let’s see. That was ’58, probably, so I was 22, yeah.

Knuth: You had one-on-one also with the IBM 650.

Feigenbaum: Yeah. Personal computer! I loved it. I loved the lights, I loved pressing the switches.

Knuth: Did you write code for IPL later on?

Feigenbaum: Yeah. So IPL-I was a paper language. IPL-II was rapidly implemented within a period of two months maybe, or three months, by Cliff Shaw at the Rand Corporation, with a lot of contribution and steering by Allen Newell. Rand had allowed them to put in a teletype connection between Al’s apartment, or Al’s house in Pittsburgh, and Cliff’s office, so they were able to exchange a lot. IPL-II came along, and in fact the logic theory program that was presented to the Dartmouth Conference -- this so-called kickoff conference for AI, the one that McCarthy had organized -- was actually written in IPL-II. It actually first ran on a computer in IPL-II. ,Then Newell and Shaw were doing experiments rapidly. They did an IPL-III, which was carrying IPL-II to an extreme. It was too extreme, so it was quickly abandoned. Then they proceeded on to a new set of ideas called IPL-IV for the Johniac. Now IPL-IV’s idea seemed to be well honed enough that they could be tuned, made a little bit better and to produce the first public list processing language for a machine that a lot of people had. I think in fact maybe in the interim there may have been a version called IPL 650 that Fred Tonge wrote based on IPL-IV, but IPL-V was the first big public language, and we did a manual. I think you even mentioned this in your oral history. You did the manual first, so that the system would conform to the manual. This is what we did with IPL-V, and then I got to do the implementation of it in the summer at Rand.

Knuth: IPL-V was running on several computers or just one?

Feigenbaum: It was running on the 709, I think, by that time. The 704 had been replaced with the 709, but that was quickly replaced by the 7090/7094, so that whole series of computers were running IPL-V.

Knuth: This was out at Rand.

Feigenbaum: No, anywhere. It was Rand where I did the writing and debugging in summers.

Knuth: The Johniac was also at Rand…

Feigenbaum: The Johniac was at Rand, yeah. The Johniac's down here now in the Computer History Museum.

Knuth: Good. I'm just wondering, as you're a graduate student, what it's like using the computers in those days? You're punching the cards. You're not submitting them to somebody else, right? You're—

Feigenbaum: No. That's right. That is one of the great things that happened at Carnegie. I don't know how it was at MIT, but at Carnegie there was no staff between you and the computer. You could book time on the computer, then you went to the computer and did your thing.

Knuth: What I mean is, was it the same at Rand? Was it the same at IBM?

Feigenbaum: No.

Knuth: Was it the same—

Feigenbaum: No. At Rand, on the Johniac, you could use the machine just by going up to it and running things. There were no such thing as operating systems. There were people called operators, and you would write out instructions to them as to what to do. What tapes to mount, here's a card deck, run it this way, if it stops do this. One of the great things about having an operator is that they're humans that you can become friends with, and you can persuade. If you are in desperate shape, if you just can't stand one run a day, you really need to stay overnight. God, these guys could run your job between everybody else's job, and give you one pass every hour, or something like that. That's what I would do -- I'd make friends with them, and stay up all night and do that.

Knuth: At Carnegie you didn't have to.

Feigenbaum: No.

Knuth: You were the operator there.

Feigenbaum: Yeah. I think it's important to tell a story. You alluded to it. Not the same story, but you alluded to the idea in your oral history and it's been very important for my career. I feel it as the experimental approach to computer science as opposed to the theoretical approach to computer science. The story is when I ended up in '59 at the National Physical Laboratory in England in Teddington. The machine that Turing had helped to design, the Pilot ACE, was still being built there, but it was running. I saw a guy sitting at the console of that a lot. I never got to meet him and I never introduced myself to him, but it turned out to be Jim Wilkinson. Later on at a lecture at Stanford -- the Forsythe Lecture at Stanford -- Wilkinson describes how he's sitting at the console at NPL and he's watching the lights flash and they are undergoing a periodicity that was unexpected to him as a numerical analyst. There shouldn't have been that periodicity in the lights moving back and forth. He says "that's how I discovered" whatever we give him such great credit for. It's really important to be in touch with the machine, as opposed to the way Maurice Wilkes is running Viatec[ph?]. My two best friends in England at that time were Roger Needham and Karen Sparck Jones. Roger's dead now -- as you know, a very famous computer scientist. Karen Sparck Jones just won the Allen Newell Award at the ACM. They wanted to know all about list processing. In fact, they really didn't know about higher level languages, because Wilkes' view was the EDSAC time was too valuable to waste it on translating languages. And furthermore, you had to shove your cards through a slot and get your paper output the next morning. Bad idea.

Knuth: I guess we're getting a little ahead, to England, there. Your first conference paper is about microeconomics, and you're helping people as the computer grew--

Feigenbaum: Yeah.

Knuth: --just like I did with Latin squares a few years later. But the main work that you're doing your thesis in is to figure out something about the way human beings work.

Feingenbaum: Yes.

Knuth: Let's talk about the EPAM -- how the idea came to you.

Feigenbaum: EPAM stands for Elementary Perceiver and Memorizer. It was a computer simulation model. Now by that I mean I'm mentioning the technique, or the methodology, of elementary information processes that we thought were going on in the mind while certain kinds of tasks were being done. There's a model of mind, and the particular tasks happen to be classical tasks that psychologists do with their hordes of undergraduates.

Knuth: For example?

Feigenbaum: Yeah, I'll tell you. These tasks were sort of -- I think they were invented by William James and Ebbinghaus back in plus or minus 1900, around there. They are tasks in the learning of nonsense verbal material. If you're trying to understand basic learning -- learning that doesn't have meaning attached to it -- what are the phenomena of that? They would have long lists of nonsense trigrams that were actually compiled and available and you can use them standard. DAX is a nonsense trigram. JIR.

Knuth: One of them was DNA, probably. It was nonsense at the time.

Feigenbaum: At the time. No, they always had to have a vowel in the middle and consonants on the end. They were actually seriously ranked by a lot of work. They were ranked in terms of how nonsensical they were. Over the years they had produced long lists of these things. These nonsense syllables were presented either in pairs, what you might call an input and an output pair. See this, say that. The order of the pairs was randomized from what's called trials, to trial, learning experience to learning experience, until the subject got it right -- all the pairs. Or the nonsense items were sequenced, so that you'd have a list of 10, or a list of 12, and there were phenomena associated with serial learning. There were phenomena associated with paired associate learning. There was certain kinds of errors that subjects would make. There was a distribution of those errors of different types, and a variety of phenomena that were well codified in the psychological literature. One of the most stable elementary pieces of the psychological literature.

Knuth: It was a repeatable experiment, and different scientists would get the same results.

Feigenbaum: Yes, and not only that. We at Carnegie Tech didn't have to do it in our laboratory. There were hundreds and thousands of those that had been done, and they were all summarized beautifully in the Handbook of Experimental Psychology by Carl Hovland, and an article which came out more or less that time. People were busy doing new things in that area too. There was a Scientific American article about it roughly at that time.

So I was fortunate in that when I got back from IBM and showed up in Herb's office in September of 1956 and said, "Here I am,", of course Herb's attitude was "You don't do courses. This is research. Here's a problem. Go think about that. How can you make sense of this literature, and what are some underlying principles, where you could pick two or three principles and it would give rise to everything -- you could explain all these results?" Maybe he didn't say this, but as it turned out it worked out that way --predict some, not just explain it. So I got to work on that. He and Al Newell were doing other things. They were doing exactly the same thing with regard to richer human problem solving. This was not a rich problem solving task. This was a perception test. You perceived characteristics of nonsense trigrams, and you learned features of them and produced output. But they were working on GPS, a General Problem Solver. their problem set -- what I might call their sandbox, their experimental domain -- was either puzzle solving or trigonometric identities. Both of those show up in their book "Human Problem Solving", which was published in 1972. that was more complex. That was Means-Ends Analysis -- much bigger program -- but both were started at about the same time.

Knuth: I'm not quite sure, but there was this logic theory machine which would just be able to tell whether a logical argument was correct. But then they wanted to say, "Now how would somebody actually construct a proof?"

Feigenbaum: Yeah. Let me unravel that. That's a real good question. Prior to GPS there was, let me say, the launching platform for essentially all of heuristic programming -- the heuristic programming mainstream of AI that lasted for 20 years or so. That was a thing called the Logic Theorist. That was what Newell and Simon were working on in the fall of 1955, and that came to a head on December 15, 1955. That was a proof-finding program, not a proof-checking program. First of all, it was not an attempt to rigorously compare the output of the program with exactly what a human would do in that situation. It was more of the "let's feel it out and we'll get to that later but right now we're just going to make it do something". On the other hand, all the insights were from what you might call bi-introspection from Herb Simon and Al Newell, of what was going on in their heads. What were the problems? The problems were the theorems of chapter 2 of Whitehead and Russell. Why did they choose Principia? It was a brilliant choice. Choose the propositional calculus, because you don't have to deal with anything having to do with the semantics of the world. You just have P's, Q's and R's, and rules for manipulating. They came up with the idea of the domain, the search for solutions in a space, and then pruning -- that was an enormous space of possibilities -- pruning that space using various kinds of heuristics. Like for example, if by going down one of these paths you happen to generate a formula which has a lot of negations in it, give it up. Because chaining back through not not P implies P. It's just too hard. It just gets you into a lot of tangles. So forget it. There's got to be another path which will get you there.

Knuth: This is one of their expert rules in order to prune off probable unfruitful lines.

Feigenbaum: Yeah, exactly.

Knuth: And they're working both forward and backward or—

Feigenbaum: No, in this particular case I think they were just working….

Knuth: Going from the hypothesis—

Feigenbaum: They had a theorem to prove, and they were going down to the axioms. Anyway. the Logic Theorist was published. Its first manifestation was in the 1956 IEEE Journal where Chomsky had his famous paper. It was done in, I believe, September after that Dartmouth conference. They brought it to the Dartmouth conference, they published it in September in the IEEE. A little bit later on they published it in, of all places, Psychological Review, which is the main kind of theoretical journal for psychology. So it was almost simultaneously published in two kinds of fields: psychology and what we now call computer science or information processing science.

Knuth: GPS went beyond this.

Feigenbaum: Yes, GPS was an attempt to pull out a more uniform mechanism and organize it in a way where it could be applied to many different problem domains. It's a trick that repeatedly we've done in AI and repeatedly I've done in my own work. For example -- I wasn't specifically involved in detail but my lab did it -- Buchanan and Shortliffe did a medical diagnosis program called Mycin. We extracted from Mycin the core of it and called it E-Mycin for Essential Mycin, or Empty Mycin, and that became the basis of hundreds of copies or thousands of copies of rule-based software.

Knuth: In other words, the Logic Theory machine is working on Whitehead and Russell. But now you could also use the same paradigms, for let's, say trigonometry.

Feigenbaum: Yes. Now, it's not exactly the same, because in GPS Newell and Simon imported from some earlier work of Herb Simon's -- earlier going way, way back to the book "Administrative Behavior" (a footnote about that book a little later) -- a thing called means-ends analysis. Working back and forth between—

Knuth: Both ways.

Feigenbaum: Both ways, right, and—

Knuth: New paradigms.

Feigenbaum: Yeah, a new paradigm. So here's where I am. And here's the goal that I've set for myself. What's the difference between where I am and the goal? Well, let's set out to reduce the differences one at a time. Now I make a step. Now I'm here and there's a goal. What's the difference between where I am and the goal? Let's reduce the difference. Finally, you get to the point where, hey, I'm really there and I have no more to do.

Knuth: Trying to find a good way to quantify the difference between two concepts or states of—

Feigenbaum: States of the problem. Now there is a critical issue here which didn't come up for Newell and Simon. It didn't come up explicitly, but I got to tell you that it was there implicitly because I saw it. I mentioned that in the Logic Theorist they avoided the question of real world knowledge because it didn't matter what the P's, Q's and R's were. Then they went on to GPS. You can't get GPS to solve a problem until you tell GPS something about the problem. The question is, how much do you have to tell it? They boiled that down to a thing called the Operator Difference Table. Meaning, if I measure a difference between where I am and where I'm going, there's some operator that's relevant to making the step and let's try that. The Operator Difference Table for, say, trigonometry that they used was really very small. I can't remember the exact number of entries in it, but I could draw it out on one Excel sheet, I think. Which means that the amount of real world knowledge that they were pumping into GPS to do trigonometry was very small. But that could be, because the trigonometric identity symbols really have very little meaning. They're a system closed in on themselves.

Knuth: They probably didn't want to start with the hardest problem.

Feigenbaum: Yeah, that's right, and then they--

Knuth: **--**with something that would give experience--

Feigenbaum: No it was more than that. It was that they were looking-- Their mindset was on problem-solving processes, not world class behavior. Not getting the job done, but what is the best model of a human doing problem solving. In fact, they do a brilliant job in their book in 1972 of matching human thinking-aloud protocols of trigonometric problem-solving with exactly what GPS is saying at each point.

Knuth: Then other people developed theories that are for machines and humans can't do it all. Simplification programs now, and your languages now use fancy algorithms—

Feigenbaum: Yes, absolutely.

Knuth: --that are completely different from the—

Feigenbaum: Yeah, and the issue you're mentioning arose immediately. Who brought it up? Wang, I think, brought it up in 1956 when they published the Logic Theorist and he just said, "This is a ridiculous way to prove theorems in propositional calculus. There's this algorithm you can use and it will do it all in five seconds." To which the answer is: we knew that, but that's not what we're out after. We're not looking for that.

Knuth: Exactly. We get into that all the time. People say, "Why do you do an experiment in order to learn something, when you can get the answers to all your toy problems by other means?" And now, "Why teach anybody to write a computer program if there is already one on the web that does it or something like this." How is anybody ever going to learn something if they don't—

Feigenbaum: Yeah. I agree.

Knuth: --if they don't know the problem solving process—

Feigenbaum: Don, let me just mention two things. One is to reinforce what you just said about experiment. Experiment turns out to be absolutely vital, at least for me and, I noticed, for Herb Simon, even though he was a great mind, and Al Newell was a great mind. Al Newell was the ultimate experimentalist. He was hands on. I will program everything just like you, just really down, but Herb would go to lectures with a big printout in front of him and while the lecture was going on he'd be debugging his code.

The other thing I want to mention is what an amazing -- what shall I call it? -- I could use an astronomical term. I can use a Kuhn-sian term, which is to say a paradigm shift or a revolution in thought. Or you can use an astronomical term, which is a supernova of thought -- was this 1956 introduction (or '56 and '57 introduction) by Simon and Newell, and then also beefed up by the EPAM results, of trying to understand thinking in concrete terms, where everything had to be explicated. Before that, [in] books on thinking, you get all the literature, [all] words. It's all English language, which means you don't confront all the issues.

Knuth: It's so easy to forget how much we don't know until we try to explain it to a computer.

Feigenbaum: Yeah, exactly right.

Knuth: What you're saying is-- To me the thing that differentiates computer scientists' ways of doing things from others most strongly is the way they jump levels, high level to low level, and they're working sort of simultaneously on all levels at once. As you say, he's debugging his program and he's listening to a lecture and so on, but you have to see things in the small and in the large. And a facility for that seems to be what brings people together. They meet other people that have that same facility, and they become friends because of that, rather than because they have a goal to do some computation they're really… The community comes from the fact that they share this talent for jumping levels.

Feigenbaum: Yeah.

Knuth: It's off the subject though.

Feigenbaum: No, it's not off the subject.

Knuth: I think it's good. We've set the scene. We see now the time when computers are starting to enter the world, and people are seeing we can actually use them to do things, are able to check things that scientists have been doing in studying human beings. Now we can actually do simulations and see if these models are at all complete, or if there's something that wasn't captured by the people who didn't have a way to check it. So here we are in a brand new world. It's very exciting. People are coming together, like at the Dartmouth conference, and seeing that there is this potential that these machines might be able to do fantastic things. Who knows what they're going to be? It was a very exciting time to live. I can also see why people might also want to make very optimistic predictions at that time as to how long it would take to translate languages, or play world class chess. At this time you are… If I understand it, then Herb says, "Here's this field of study with these nonsense,…

Feigenbaum: Yeah.

Knuth: …what do we have to do in order to validate these experiments, by actually implementing them precisely and getting to the bottom of it?"

Feigenbaum: Yeah, right. There were some clues lying around. You have to sort of pick up breadcrumbs in the forest. One of the clues was, in the post World War II era of cyberneticians there were some people in the MIT area that came from psychology. George Miller being one of them, Licklider being another one. There were several others who are—

Knuth: McCulloch—

Feigenbaum: McCulloch was actually a psychiatrist, and he has speculated about neuromechanisms. But these other guys were actually working on psychological things. Like: Miller had this famous paper called "The Magical Number Seven Plus or Minus Two", in which he did experiments on the span of immediate attention. Well, there were some that had to do with-- Oh, Neisser was another one, Ulric Neisser. Well, some of them had to do with choice reaction time, and it turned out to be that the reaction time was related to the log of the number of choices. Once you see that, you know that some [kind] of a tree is happening underneath. So the first thing I invented was a thing called the Discrimination Net, which started out as nothing when the learner starts. Then the learner, under an assumption that when-- It's kind of an efficiency assumption, similar to what a physicist might make: [take the] path of least energy, of doing the least amount. Just differentiate what you have to differentiate. If you have a nonsense syllable like DAX and one like JAR, it's well known from other results that people will discriminate on the ends of those before they discriminate out of the middle. So just discriminate between the "D" and the "J." That's all you have to do. And leave a little cue around that lets you-- The "D" for example-- Well, no, I can't say-- Yeah. The "D" is a closed loop and the "J" is not so that's- there is a feature that you can discriminate on.

Well, it turns out that list structures had just been invented by us, and no one had tried to grow these trees. But I had to, because I would start with two nonsense syllables in the net, and then the next pair would come in and they'd have to grow into the net somewhere. So these were the first growing trees, first adaptively growing trees. Now here's the amazing and kind of stupid thing, but it just shows what it means to focus your attention on X rather than Y: we were focused on psychology. We were not focused on what is now called computer science. So we never really published anything about those trees, except as they related to the psychological model. But other people saw trees as a thing to write papers about.

Knuth: I think Ed Fredkin.

Feigenbaum: Ed Fredkin's "tree memory", for example. Yeah. I might have become well known in some other area if I had just thought of it, but we weren't thinking that way.

Knuth: From a historical standpoint, trees were sort of downplayed in all of mathematics and so on, until computers came along when we started having things like recursion and with branching. You started out with IPL-V. IPL-V has lists that are one-dimensional. The next step is to have two links, where you can go two ways instead of one. It seems obvious in retrospect, but really it wasn't a common model at all in people's minds, and it wasn't until later that we saw, really, that trees were everywhere.

END OF TAPE 2

Feigenbaum: Don, I'd like to pick up on a couple loose ends that we were talking about, [and] just make some comments on them before I forget them. One is about list structures. I actually can't tell you for sure about IPL-II, but [in] the list structures of IPL-V, of course, every list structure had a name, so that's the object name. It represented an object. But if the object had structure, it was either a list, or a list of lists of lists of lists of lists. Namely, a structure. But if aside from the structure, it also had attributes, and those attributes had values, there was what's called a description list that was hanging off the name of the object: attribute-value, attribute-value, attribute-value, attribute-value. "The color is red." "The size is large." Something like that. Well, that was heavily used as a programming mechanism by us -- by Newell and Simon and myself and others. Years later it was reinvented, and it was called "object oriented programming". But object oriented programming was invented right there, in doing that! That program. Iit was [later] made pretty, and it was made public, and all that. It was just like: McCarthy gave a name to a search technique, a pruning technique called alpha beta, which of course you've written a lot about. But when McCarthy first did that, Art Samuel and Al Newell looked at it and said, "Well, of course.

But you wouldn't program any other way. This is silly to call it. We've been doing it." It's like, "I've been writing prose my whole life".

Knuth: I would add [that] object oriented programming also included the idea of an executable procedure as part of the object.

Feigenbaum: We also had that. Any one of those symbols could stand for a program that you just jumped into the interpretative stack and it got done. Incidentally, that's a good point: these were interpreters, not compilers.

Knuth: Right.

Feigenbaum: It was way later before there was a compiler. I think even early list was strictly interpretative.

Knuth: I'm willing to predict that in the next 20 years interpreters will be rediscovered at least three times too, as to how important they are.

Well, let me see. There's so many things to go into. With respect to EPAM, my main question is, more clearly: it was a success that you were able to find this good model But my question is, now will people continue doing brain research for another 40 years? Does it correspond now to the way today's researchers think the brain works?

Feigenbaum: Really good question. Really good question. This tape, this video, will be watched by people 50 years from now, 100 years from now, who know the answer to that. So anything I say right now is going to look silly to them. But I just have to say what I think about that, because you've asked a super important question.

The current run of brain studies, which are based largely on what's called FMRI data, Functional MRI data, don't answer the question in any detail at all. They just talk about body energy being burned in one area of the brain versus another area of the brain. There's not a lot of detail involved in that. In physics there are intermediate levels of theorizing that sit between what you might call the lowest or ultimate levels of theory building and the big picture. For example, when in college physics we studied "pV =nRT," we know that p, pressure, is not a fundamental in the world, it's a result of molecules banging up against a surface. Which means that it has something to do with molecules and statistical mechanics underneath. There's this intermediate level of theorizing. Well, all of us who were working on these things like EPAM and GPS and so on, and for a long time what was considered the mainstream of AI, were working on a level of theorizing which I call the level of the mind, not the brain. Therefore, the term "giant brains," like Ed Berkeley did, was just offensive to those of us who were working in AI, because we were talking about symbolic processing in minds, not brains. In fact, I even thought that it would be forever before there would be a connection between symbols and neurons and neural systems. When my daughter got to Oxford to study -- she was in the neuropsychology field program at Oxford -- they thought because her last name was Feigenbaum that she knew something about neuropsychology and

AI. They would ask her questions like this. But actually people like us -- Newell and myself and Simon, and 50 other people, Minsky and lots of others, even John Anderson, who was directly in psychology, he won the big prize in psychology, he was a student of Gordon Bowers and mine at Stanford -- they were working at the mind level. Like associative networks. John Anderson's thesis was HAN, Human Associative Network.

People recently have gotten enchanted with the idea that you could actually find out something about how the brain works. But it's very, very early in that, and they're making wild guesses and tremendous jumps without filling in the details. For example, in Newell and Simon's book "Human Problem Solving" they make a very strong case for a certain set of what they call EIPs, Elementary Information Processes. And yet there has been no attempt to link EIPs with what's going on in the brain. If they could, then they could make a big jump between the brain and problem solving.

Knuth: Exactly. It seems to me that even your 50 years from now might be an optimistic estimate as to whether people are going to understand cognition. It's such a mystery. But the more hardware we get, the more MRIs we get, the more tempted we are to think we're almost there. But, as you say, it's going to be a while before people know. But as far as anybody knows, the EPAM works. It still has some kind of physical manifestation, but we can't nail it down saying here's where--

Feigenbaum: Can't nail it down into brains, yeah.

Knuth: Yeah. That's good, thank you. Another question before I forget: I want to know something about working style. You mentioned Newell would do a lot of programming and so on. Did they do a lot of their work, like Newell and Simon, or you and Simon and so on, in the same room over a period of time, brainstorming? Or was it that you'd meet each other weekly, and then go off and work intensively on your own? Or what?

Feigenbaum: Good question. For me it was essentially -- working on EPAM, anyway -- it was working on my own during the week, and then making sure that I got in a fairly long session with Herb Simon every Saturday morning. Herb would put away Saturday as the day to do sustained research and write papers. So Saturday morning I could get some good time with Herb. But if odd things happened during the week, whether it was good or bad, and I needed to talk to Herb to tell him something… Since we were in the graduate school of Industrial Administration, the culture was a mix-up culture. There was a very important event that happened every day at three o'clock, where everybody, no matter what they were doing, just came down to the lounge and had coffee or tea.

Knuth: When you say everybody, this is 30 people?

Feigenbaum: Yeah. 40 people. The macroeconomists, the students, the graduate students. Not the undergrads or the master's students. The research-oriented people would just flood the commons room. Occasionally we'd have speakers come in from the outside, friends of Simon's or Bach's or whoever. George Stantis <ph?> might show up and talk about his thing.

Knuth: Were you able to recreate that later on, when you were running labs?

Feigenbaum: We never did it in Computer Science [at Stanford]. We talked about it a lot with Bob Floyd when the building was being planned, about spaces in which that could happen, and it never quite came off. In the new building, really nothing like that happens.

Knuth: Well, maybe people hearing this oral history will pick up on it then again, because it certainly worked. We know a lot of examples where it worked. But you're right, now people are much less personal, less interactive between the different flavors, as things are getting more complicated.

Feigenbaum: Just a footnote to what you just asked: Newell would see Simon. Newel had more access to Herb than I did, because he's more senior than I was -- way more senior. But Newell worked mostly at home and on his teletype with Cliff Shaw. He would come in a couple times a week to talk to Herb, but he wouldn't show up at these three o'clock things because it was an extra trip for him from home.

Knuth: He was really on a teletype to Los Angeles?

Feigenbaum: Yeah, Rand teletype. Person-to-person teletype.

Knuth: I was in Cleveland, Ohio, at that time, but I never heard of anybody communicating with another [like that]. Except you'd send telegraphs -- you know, when you'd get these funny paper things. But that's wonderful.

Feigenbaum: Well, remember they were operating on Air Force money. Rand was a big deal Air Force contract.

Knuth: Okay. Because they didn't have timesharing at MIT until a lot later.

Feigenbaum: Yeah, this was person-to-person. This wasn't computer–to-computer.

Knuth: Okay, very good. Now I guess we should be moving into why you get to Cambridge during 1959.

Feigenbaum: I wanna back up just one second before you do that, to say two last things about EPAM. The first is that it was the source of my first significant "aha" moment in my whole scientific life, which is the following: Herb and I were working on the assumptions that we were building into this learning model -- and I want to emphasize learning model -- acquisition, putting things into memory, building branches of this tree. I was running experiments. I would literally run them on the computer. I would literally be sitting on the 650, and I'd run in cards, and they were just simulating a psychological experiment. I was literally running it. Then one day I'm running it. I'm running some of these. We wanted minimal assumptions. We didn't want a bell and whistle theory, so minimal assumptions. We're pruning everything to the minimal, and I'm putting this stuff in. And this program, this simulation, is learning stuff! Trial 1; trial 2; trial 3; trial 4. It forgets it; trial 5, it's back; trial 6, it's gone; 7, 8, it's there; 9, it's gone. What is going on? I mean, is that a bug in the program or ?? Well, it turns out it's not a bug in the program. It's this minimalist assumption that you never should store more than you need for access at any moment. In other words, recognition is different than recall. If you want to access something in a recognition state, recognition mode, you store a small amount of information about it. But the more you learn the less adequate becomes that information. All of a sudden, the access to it disappears. Well, you have to relearn that by adding a little more information. And you only need to put in a little more information and you can relearn it in the very same trial, so it comes back right away. Well, it turns out that's an exact match to the psychological phenomena. That you do have this thing called "interference forgetting". The marvelous thing about interference forgetting is they come back right away. Then they disappear again, and they come back right away. Tthat was an amazing thing, because I never put in any assumptions about forgetting at all. It just dropped out of the theory, which I loved.

Knuth: Exactly. That's great.

Feigenbaum: And the second thing I want to say about EPAM is that Herb worked on it until the time he died. It was on his list. He had a list of the 10 books he was working on, literally, and on that list was EPAM. He had still been working on it with a student named Howard Richman. He and Howard had published a lot of papers. They had programmed IPL-V for the PC. By that time, they had evolved [it] into EPAM6 and it was running on the PC via IPL-V, and it got very more extensive. But by the time it was finished -- well, even when I left it I didn't do much more work on it after 1965 -- but even when I left it, it was the best going theory in that verbal learning area that there was in psychology. It still is.

Knuth: Is there a Java applet? We can't easily just try it out on some website right now?

Feigenbaum: No. I actually just recently donated some floppy disks to the Computer History Museum that I think had EPAM running on IPL-V for the PC, but I could be wrong. But we can get it from Howard Richman.

Knuth: There are ways now to simulate simulators of the past, which is great. Then you got your thesis in hand but before you graduated you spent a year abroad.

Feigenbaum: I finished my orals on some morning in September in 1959. Herb's view was a thesis should take three years. I tried to get that through at Stanford. That happened with Doug Lenat, and it happened with Ted Shortliffe, but very few others finished in three years . I took my orals on some September morning in 1959 and flew to New York. In the afternoon I got on the SS United States, the boat, to go to England on my Fulbright. But Carnegie Mellon didn't award degrees in September, they only awarded them in June. Although the work was done, the degree was awarded in June of 1960. Meanwhile I was off in England, and I never showed up for my actual graduation because I was off and rushing.

Knuth: I was wondering about that, because I knew airplane travel was not very common to go across the Atlantic at that time. The Fulbright program is only about 10 years old at that time. Are there lots and lots-- is it already a real big program, or are there very few Americans who were getting Fulbrights?

Feigenbaum: Let me give that some thought. I'm guessing to say that the number of Americans that showed up going to England was, this is a rough guess, I'll say 20, plus or minus 10.

Knuth: This would be in all fields?

Feigenbaum: Oh, yeah, yeah. In fact, it's interesting, one of my friends at the time, Nancy Andreasen, who was an English major at the time, was going to do English at Oxford. She ended up being one of these neuroscientists that we're talking about. She's written books on the brain in her subsequent academic career. And one of my friends from Carnegie Mellon was a Fulbright scholar, Dave Hummer. He ended up heading the joint laboratory of the astrophysics lab at Colorado.

Knuth: Did you have a choice of many places to visit?

Feigenbaum: You mean to stay at? I could have had my pick, but I picked NPL because, as they would have said in Casablanca, I was misinformed. NPL, National Physical Lab, had hosted what essentially, except for the Dartmouth Workshop, was the first big public AI conference, [in] 1958. It was called the Mechanization of Thought Processes. Butley, Alfred Butley <ph?> was the director of the lab and he ran the conference. I took that as a sign that they were active in this area, and that's why I chose them. And it was in the London area, and that would satisfy my need to be around concerts and plays, etc. I didn't even think of Cambridge. But, actually, the real good stuff in the country was going on at Cambridge in Manchester, not at NPL. NPL was great, I guess, when Turing was there.

Knuth: What about, like, Edinburgh [Scotland]?

Feigenbaum: Oh, that came way later. That was Donald Michie's creation.

Knuth: I see. Much later.

Feigenbaum: Much later. That was when Donald Michie, who started out his academic life as a geneticist, was working for Turing in Bletchley Park and worked on Colossus. That's when he gradually made this transition into AI and eventually showed up at Edinburgh to start a lab.

Knuth: I've never been to NPL, but you say it's near London?

Feigenbaum: Yeah. It's about 10 miles beyond the end of the Green Line. The Green Line ends at Richmond, and you keep going in the direction of the Hampton Court Palace and you go right to Teddington.

Knuth: Oh yeah. Then of course, Cambridge is to the north a ways. So you didn't spend much time at NPL but you went up--

Feigenbaum: I'd go there every day. I bought myself a little Volkswagen. Picked [it] up at the factory in Germany for the amazing price of eleven hundred dollars, can you believe? Brand new Volkswagen. I would drive down from Richmond, where I lived. I lived on the end of a line so I could take the train into London when I wanted to, and drive to the lab when I needed to. I went there every day. But I can describe to you in great detail why it was so dull. I know it isn't, though, because Wilkinson did his wonderful work there, and Donald Davies did that incredible work that he never gets credit for on the birth of packet switching and internet ideas. We Americans just never give Davies credit, but he did it there. But it seemed really, really, really dull. When I got there at NPL, having come from an environment where I was programming on the 709 and we had everything we needed in terms of printers and card punches and blah,blah,blah, they still were punching cards on what we would call binary correction, one at a time. If you had to do a character, and you needed two punches, you had this one card in there and you had to move it there and punch it there and move it there and punch it there. This is ridiculous. This is 1959!

Knuth: And submit it to the operator, as you said.

Feigenbaum: So, I happened to make contact with -- I think the original contact came from Roger Needham, although it may have come from his wife, Karen Spark-Jones, but anyway, they knew that I knew about list processing. They also knew that I knew about compilers because I had come from Perlis' place. They wanted to make contact with an American who knew about these things, so the first time we met was when Roger and Karen came down from Cambridge to NPL. They were such great people. Karen introduced me to Margaret Masterman, who was the head of her language research unit. Margaret introduced me to her husband, who was a philosopher. It was a great scene. I spent a lot of time with them.

Knuth: There's this atomic energy lab. Is that not connected with--

Feigenbaum: Nothing.

Knuth: No connection? Because that's where Glennie was. He was doing compilers there. You never met Alec Glennie?

Feigenbaum: No.

Knuth: By the way, I can say that conditions were even worse at Cal Tech in 1960 when I got there as a graduate student. I don't think we had punch cards. We had paper tape. It was really hard to use the computers there. It was like that all around the world, in spite of the fact that you're spoiled by having worked at Rand and so on. All right. You told me once that you wished you hadn't graduated so soon?

Feigenbaum: Oh, yeah. Good point. I spent a year in England, and the highpoint of my year was meeting Karen and Roger and having a good time traveling around the continent, etc..

Knuth: I just want to get it straight -- you also went to Cambridge a few times?

Feigenbaum: Yeah, sure.

Knuth: But not very often, ok?

Feigenbaum: Right. I would either drive there or take the train up. Mostly there was just more there, but for Roger and Karen coming down to NPL. I could see a lot of other people there. I realized later in life that what I needed to learn from Herb Simon-- now we're talking about one of the great scholars of all time, and a genius. And I was just lucky enough, by pure accident, to get into orbit around Herb. It was just stupid to leave after three years, even though Herb thought a three year degree is what you should get. He offered me a post doc. I could have stayed there as a post doc. Actually, I think the offer was like a two-year post doc. But I sort of wanted to see the world. I hadn't been out of the United States except for a little trip to Canada. I had gotten married by that time. My wife wanted to travel also, and it didn't seem like what I wanted to do was just do more sitting around doing EPAM. So we traveled. But in retrospect, I just had a lot more to learn from Herb about the processes of doing science -- what I call the heuristics of doing science. That's why I strongly believe that the Ph.D. is a research training apprenticeship, and you have to be around the person doing the research and absorb what they do.

Knuth: But you came out okay. What would you really… You would have your life on hold for another year, and you think it would have been even better?

Feigenbaum: I think it would have been better to skip the England experience and spend it with Herb.

Knuth: Oh, but the international, the idea… I mean, later on you become a world traveler, you make all these connections, you're building up more paradigms in your head for learning about organizations and different lifestyles. You have to say, "Well, if I had more of this, what would I take away?"

Feigenbaum: That's right.

Knuth: It's good to get your gut feeling about this. Of course there's all kinds of things in our life -- we wish we had another year to do this, and another year to do that. You decided not to do a post doc because you still liked variety and challenges of new things. So now your first job…

Feigenbaum: I had arranged my first job before going off to England. I knew what it was going to be. After high school I had traveled around the country and decided, oh, San Francisco Bay Area -- that's where I want to live. That meant I had two choices, Berkeley or Stanford.

Knuth: No, sorry -- the first choice is academia or not academia.

Feigenbaum: Oh, yeah. Actually--

Knuth: So somehow you decided you also wanted to be involved in teaching.

Feigenbaum: Okay, good point. Don, you are so smart. In 1956, when my parents were still thinking I'd be an engineer, go out, get a job, and earn some money, I actually interviewed at Raytheon in Palo Alto. There is no Raytheon now. It was on the other side of Bayshore [Freeway], and there was a big plant, Raytheon. But I knew I wasn't going to do that. I knew I was going to be a graduate student. It was already spring, and I had already decided to go academic. Well, as an academic you could go to a research lab, or you could go to a university. I guess I even saw in Al Newell a person who had chosen research lab at that time. Newell eventually switched from Rand to Carnegie Mellon. But Herb Simon was such a strong influence that it literally didn't enter my mind to go to a research lab, so I never even searched for that. When I was searching for a job, as usual, you go to your advisor. I told him I wanted to go to the Bay Area. Could he work anything at Berkeley or Stanford? The first thing he did was to say, "Stanford." He wrote to Pat Suppes and said, "Pat, I have this great student, can you get him into the psychology department? That's where he belongs." Pat wrote back and said, "No, we don't want him." More or less -- I never saw that exchange of correspondence.

But there were people who loved me, and that was at a new research group that was being started in the Berkeley business school. They had a kind of Carnegie Tech envy. They were going to bring in mathematical… They had one guy who… When I did that economic simulation with March and Cyert we thought -- Cyert, March and I thought -- we were the first, But it turned out we were not. We were the second. The guy who was the first was Austin Hogg <ph?>. He had done it at IBM, but he had ended up at this new lab at Berkeley. Then Wes Churchman, the operations research guy -- who was also a philosopher -- was also interested in models of mind. Because of his operations research background, he ended up at that lab. Another mathematical economist named Fred Balderston, who later became one of the vice presidents of the statewide universities of California, was there. And they really wanted us. They really wanted people who represented that Herb Simon flow. So they offered both of us jobs.

Knuth: They're running a graduate program?

Feigenbaum: Yeah, and a master's program, and an undergraduate.

Knuth: How many people?

Feigenbaum: Big business school.

Knuth: FTE's?

Feigenbaum: Oh, I don't know. That was a big business school. But in the management science lab, maybe-- Hogg and Balderston, Churchman, Julian and myself -- maybe six FTE's, all told.

Knuth: Julian Feldman was already there?

Feigenbaum: Yeah, Julian Feldman. He didn't go to England. He didn't do anything except go there and get a job. He had a family. He had to earn money.

Knuth: What kind of computers did they have at Berkeley?

Feigenbaum: By that time, they had a 7090, I think.

Knuth: This was?

Feigenbaum: I got there in September, 1960. I believe it was a 7090 by that time. I think the 709 era had ended. It was over in the math building.

Knuth: Yeah, I guess Lehmer inherited that machine.

Feigenbaum: That's right.

Knuth: In later years. So, did they allow you to touch the machine?

Feigenbaum: No, no. But something happened right away that was -- well, not right away, but within that period of time when I was at Berkeley -- something miraculous happened, and it's called the [Digital Equipment Corporation] PDP-5. All of a sudden there was a thing called a minicomputer, and it was cheap enough that people with research grants could actually buy one and just stick it in your lab. So toward the end of my stay at Berkeley we actually had our own computer.

Knuth: The PDP-5 already by that time? Not [PDP]-1? Is that right?

Feigenbaum: We never could afford… I heard of the PDP-1, but that PDP-1 was like a hundred thousand dollar computer. I think the PDP-5 was down in the $35,000 range -- something like that.

Knuth: I see. Which is still a huge amount.

Feigenbaum: But anyway, computing was not a great experience at Berkeley.

Knuth: The faculty salary would be something like $10,000?

Feigenbaum: No. At Berkeley they started me at assistant professor step two. The salary level was not high enough at step one, so they put me in a step two, which sort of saved me two years on the tenure track. It was $7,600 for a nine-month year, and then I would work the grants in the summer.

Knuth: So you're teaching graduate students, or mostly doing research there?

Feigenbaum: No, I was teaching. I had a full load, whatever the full load credits for research grants that you get. I had one grant from a private foundation that Herb Simon had recommended. The private foundation had come to Herb saying we want to get into the area you're doing, who are the smart people around the country? Herb told them about me and Julian, and they gave us some money. Then Licklider [head of IPTO at ARPA] came around in '63, or something like that, basically on the same route. Talked to Herb and Al -- he knew them from way back because of the psychology connections. He came there and said, "Okay, in Berkeley I want to sponsor AI research and I want to give that money to Feigenbaum. I want to do a medium-sized timesharing machine that is not PDP-1 timesharing, and not Project MAC, but something in the middle." It turned out to be we bought the SDS-930. "I'll give that money to Harry Huskey." Harry and I were the co-PI's of that project.

Knuth: Was Harry in the same department?

Feigenbaum: No, Harry was in EE.

Knuth: He was an electrical engineer, okay. We might as well, while I'm thinking about it -- you and Harry were the co-advisors to Niklaus Wirth?

Feigenbaum: Yes. Harry was the main advisor and I was the second reader. I'm very proud of that. Nobody knows that.

Knuth: Well, that's why I wanted to bring it up.

Feigenbaum: That's why I want to go photograph Klaus' front page. There were a lot of discussions going on between [Stanford professor George] Forsythe and us, us partly because he knew Harry really well, and partly because he was interested in getting me at Stanford via McCarthy's recommendation.

Knuth: You just answered my next question. Because George was really… In my opinion, he saw that we were going to have, some day, computer science departments, and not just branches of economics departments and mathematics departments. So he was finding the best talent all over the world. I figured that he might not have pestered you while you were in England, but he probably did as soon as you got to Berkeley.

Feigenbaum: Well, actually, he didn't. It wasn't until 1964. My moving from Berkeley was a function of two things. There was a push and a pull. The pull was, because of that early coming in at step two, I was coming up very early for tenure. The business school faculty as a whole -- it was a big faculty, maybe 50 people all told -- they couldn't figure out what to make of a guy who is publishing papers in computer journals, artificial intelligence, psychology, is a member of the Institute for Human Learning at Berkeley. What is this guy? Is he a business guy? So they hemmed and hawed on the promotion to tenure. By that I mean months, instead of weeks. That triggered me to search, look around -- what are my alternatives? Well, at that point I had to make a real life decision. A major life decision. Am I going to search in psychology, or am I going to search in computer science? I did both. And while I was doing the search in computer science, McCarthy-- that's the pull, is McCarthy -- I knew John through AI stuff and AI work, and John told Forsythe that there was this good guy across the Bay.

Knuth: I see. I guess I misread that timeline a little bit there. But you did get the associate professor at Berkeley?

Feigenbaum: Yeah, I did.

Knuth: And you were 27?

Feigenbaum: By the summer of '64, I was associate professor there, so whatever that was, 28.

Knuth: Yeah, 28. So that's still counts for a little bit of hesitation on your part.

Feigenbaum: Yeah.

Knuth: But you were advising people like Wirth, who were doing --- his thesis was about compilers, so you were also training students in computer science, even though--

Feigenbaum: I had way more fun than that, because Harry took off just when we got our money. Harry took off for India on a year's sabbatical, and Dave [Evans] had just come to Berkeley. Harry hired him from Bendix. He'd just become a faculty member at Berkeley. Dave took over from Harry, and I learned more computer engineering from Dave Evans in one year -- since I was PI I got to sit in on all these meetings and help pick which machine we're going to go down the road--

Knuth: This was Evans and Sutherland?

Feigenbaum: Yeah, Dave Evans, the designer for the Bendix G-20. Fantastic engineer. I learned what engineering was all about. How you really, could actually compute things--

Knuth: So are you in Cory Hall, or are you in the same building?

Feigenbaum: No, the business school was in a historic, wonderful old Victorian building called South Hall. My office was above Wheeler Auditorium, if you know that.

Knuth: Okay.

Feigenbaum: Those guys were over in Cory.

Knuth: Okay. Good. It's time to change the tape.

END OF TAPE 3

Knuth: At Berkeley I know that you introduced at least one new course in their curriculum, and that was something like Artificial Intelligence, plus the other ones. But can you tell something about the teaching at Berkeley that was most innovative?

Feigenbaum: Most innovative? Wow, okay. Just two things occur to me about the teaching process that Julian and I were using there, and it could be that I inherited it partly from Herb Simon, or I am not sure. But anyway, there was no established literature of the field of AI at the time. But there were a couple dozen, maybe twice that number, of illuminating things to read. And some of those Julian and I…

Knuth: Julian Feldman. Sorry, this is Julian Feldman.

Feigenbaum: Julian Feldman, yes, who was a graduate student colleague of mine at Carnegie Tech and came to teach at Berkley. Jerry Feldman was another colleague of mine here at Stanford for a while, and everyone confuses Jerry with Julian.

Knuth: Right.

Feigenbaum: So we were teaching from selected papers from the literature -- like, for example, that Logic Theorist paper I was telling you about before, that was in the Tripoli proceedings of information Theory. Or my paper on EPAM -- that was from the Western Joint Computer Conference. These things were all over the place. They were appearing in a broad array of publications, media journals, and so on.

Knuth: Right. The ideas came out before their time, so there was not a natural place to publish any of them.

Feigenbaum: Right, right.

Knuth: You are co-teaching this with him as a course? Is it an elective course, or is it a seminar series course?

Feigenbaum: Oh, it was surely an elective course. I mean, I do not remember the details of it, but I cannot imagine in a business school it would be anything else but that. But it was offered to all of Berkeley, not just to the business school, because that is where I was going to get my best students -- were going to come from the math department or EE. All these papers were -- I guess we copied them -- put them on loan at the library. The students would run off to the library and read the papers. I guess maybe they could take them out overnight or something, but they did not have a lot of time to spend with them.

Julian and I decided that, gee, the time was right for a book that would bring all this together. Really writing a book is hard work and takes a long time, as you have demonstrated in your life. What we did was a collection. An edited collection. These were the papers we were using in our courses, except for a few oddities -- like there was one paper in that book on what the Soviets were calling cybernetics, but was really computer science. That was only because it was a hot topic of the time, not because it was something I was teaching in the course. But most of it was what we were teaching in the course.

Knuth: If my memory is correct, Xerox machines were quite rare at that time.

Feigenbaum: I do not know how we did that.

Knuth: At Cal Tech I had to go several buildings away to use a Xerox machine once in awhile, I mean like 1962 or so.

Feigenbaum: Yeah, you are right. And I don’t remember. When I said “copy machine” I was really nervous about saying that, because I wasn’t sure that there were even copy machines at that time.

Knuth: Yeah, right. We had these chemical things that smelled funny.

Feigenbaum: Yeah, yeah. So I don’t know exactly how the library made the copies of it. We put this book together with an introduction, which I wrote. Actually that introduction has been very… It has had a long life in AI, and it is referred to a lot by people that -- gee, that kind of thing, what you said there really brought me into the field. Then it had interstitial material between the papers to kind of tie them together so that the student would know how one led to the other, or what relationship one of these papers had with another. We took this manuscript, we sent it to the editor of the Prentice-Hall series of something. Why Prentice-Hall? First of all, Prentice-Hall was used to publishing things for people in business schools. But more important than that, the editor of the series was Herb Simon.

Knuth: Oh!

Feigenbaum: Okay. So we sent it to Herb Simon and said, “Publish this.” And he advised Prentice-Hall not to do it, that there would not be a market for it!

Knuth: Oh. And he knows everything about economics, right?

Feigenbaum: Yeah.

Knuth: A Nobel Laureate.

Feigenbaum: Like Einstein said about the cosmological constant, Herb used to say that was the worst mistake of his career. But anyway, it was published and it was highly influential.

Knuth: But it was published by Prentice-Hall or McGraw-Hill?

Feigenbaum: No. It was published by another publisher, McGraw-Hill. And then I subsequently became the editor of the McGraw-Hill computer science series.

Knuth: Yeah. This book is, I do not know if we have mentioned it, “Computers and Thought.”

Feigenbaum: “Computers and Thought” is the name of the book.

Knuth: It is not called “Computers as Brains.”

Feigenbaum: No, no, no, no!

Knuth: You were saying you did not like that.

Feigenbaum: That is right.

Knuth: The people characterizing them as giant brains.

Feigenbaum: We were modeling thinking. We were not modeling brains.

Knuth: Exactly.

Feigenbaum: Now what else do I want to say about “Computers and Thought?” Because it was a collection, it became a kind of nightmare as far as intellectual property rights and royalties. We made a proposal to the authors, Newell, Simon, Selfridge, Minski, etc., etc., etc. We said, look, no one is going to get very much money out of this book if we distribute the royalties. How about we take all the royalties and make a prize? We establish the “Computers and Thought” Award of IJCAI [International Joint Conference on Artificial Intelligence]. It is still given today. We established it as being for young people. At one point it was kind of like the best thesis in the last two years. Winograd had won that, for example, for his SHRDLU work. Lenat won it for his work.

Knuth: 1963 forward, was there an association for artificial intelligence?

Feigenbaum: No. It was just beginning an international conference called “IJCAI” that took place every two years.

Knuth: Oh, I see. The conference has some kind of a bank account.

Feigenbaum: Yeah.

Knuth: Organized as an entity. It is a very foresighted idea, I must say. So as you and Stuart are working on it together, how do you share the work?

Feigenbaum: Julian and I?

Knuth: I mean Julian. What did I say, Stuart?

Feigenbaum: I do not know what you said.

Knuth: I am thinking Stuart Feldman. I am thinking cognitive psychology. I am sorry. I am all confused here with my cognition.

Feigenbaum: I have only a misty recollection of it, except that I volunteered to write that introduction because I think both of us felt that I was more inclined to do writing than he was. And that I would be able to cover the AI side of things, as opposed to the human cognition end of things. So I wrote that introduction, and I guess maybe Julian wrote those interstitial material items and made sure that the book was packaged right. But the person who really did the work, as it turns out, was a person you know real well, Joe Traub’s wife, Pamela McCorduck. She was my secretary at the time -- Julian’s and my secretary.

Knuth: She was at Berkeley then.

Feigenbaum: She was at Berkeley. She was a junior.

Knuth: She came to study.

Feigenbaum: We hired here when she was a junior in undergraduate.

Knuth: Oh, okay.

Feigenbaum: She worked for us straight through. Then she went to Bart [?] for a while, and then came to Stanford to work for me here, and then went to graduate school in writing.

Knuth: And wrote novels like this person that you were working with at Carnegie. Okay. So that is your first of what would turn out to be quite a few books in the future years. Now let's go onto the next phase of your life. You got tenure at Berkeley in the Business School, but then at the same time you started looking around. You said you had a choice point, but you had an idea that computer science would be an existing entity of the future to be part of, or something like that. Did you say you had a choice?

Feigenbaum: Yeah, I did. Don, let me explain. The choice was: do I want to be a psychologist for the rest of my life, or do I want to be a computer scientist? By that time it was that starkly clear. I had been a member of the Institute for Human Learning at Berkeley. Leo Postman was the director, a famous experimental psychologist. Art Jensen was a member of that group -- he was the guy who provoked a lot of controversy with his book on IQ, if you remember back in the '60s. I could have chosen to go down that route. For example, of the several routes that Herb Simon chose in his life, one was economics, and he won a Nobel Prize for it. But one was psychology, and he won that same equivalent prize in psychology, and so did Newell. I could have been one of those. But for two reasons I decided… And I got an offer from the University of Oregon Psychology Department, very good one actually. I think there were maybe three reasons why I chose computer science. One was that it allowed me to remain in the San Francisco Bay area. I will give them sort of in reverse order. Second one was that by 1964, I had become fed up with psychologists. I had become fed up with their methodology, with their way of doing business. Now that does not mean Gordon Bauer at Stanford, or Dick Atkinson, or Pat Suppes, but means the general run of psychologists, just did not have a… I saw their experiments as being a whole lot of knot twiddling. They would come up with some paradigm, and they would set up an experimental paradigm, and they would farm out to three graduate students this setting of the knob, this setting of the knob, this setting of the knob, this setting of the knob. Go do that experiment.

Knuth: Okay. It had gotten away from its roots, become baroque.

Feigenbaum: Yeah, yeah.

Knuth: Institutionalized.

Feigenbaum: Right, right. But the most important reason was that I really looked inside myself, and I knew that I was a kind of a techno-geek. I loved computers, and I loved gadgets, and I loved programming, and so the dominant thread was not going to be what humans do, it was going to be what can I make computers do.

Knuth: Right. What you loved to do. Could you have stayed at Berkeley and been a computer scientist?

Feigenbaum: Yeah. I don't know if I mentioned it in this interview, but [at] Berkeley the environment was terrible for computer science in '64, because there was raging battle going on under the scenes between the Humanities and Sciences College -- I'm sorry, called Letters and Sciences -- and Engineering, as to who was going to own this new discipline. It was not pleasant. But later on it got all cleared up, and it became a very nice place. And I had tenure Berkeley. If I had just stayed there, I would not have stayed in the Business School. I would have moved over into this EECS department, or probably would have moved over to one of the two departments that actually got formed. And it would have been a great life. I had a wonderful house there. It eventually burned in that Oakland fire. Remember that huge fire?

Knuth: Oh, the Oakland fire. Oh, okay. The free speech movement was going on, starting about that time.

Feigenbaum: Yeah. That was ugly. The free speech movement was ugly. It actually was bubbling during the time. There was a period when I had already decided to go to Stanford, and I knew when I was going to leave Berkeley, which was December 31st, 1964. The free speech thing bubbled up right up in the middle, while I was on a 1964 trip to the Soviet Union. I read about the riots in Berkeley in an English language newspaper in Leningrad, which was mostly a propaganda sheet, except it talked about…it showed a picture of this guy on a car. What was his name, the famous guy? He and Joan Baez were the two big names in the Berkeley rebellion. He is the guy who got on a car. Anyway, it was going on when I got back to Berkeley after the trip. Then there were meetings of the Academic Senate, and it was pretty uncomfortable.

Knuth: You said something about being in Wheeler Hall? That is where a lot of the confrontations were, right around there too, I think.

Feigenbaum: My office was on the second floor of Wheeler. Wheeler Auditorium was on the ground floor, and my office was on the second floor.

Knuth: Okay. Well, I am so glad you chose Stanford.

Feigenbaum: Thanks to McCarthy, incidentally. That is a historical note.

Knuth: You mean John was much more of a tempting force than John Forsythe?

Feigenbaum: Well, no. But Forsythe would not have known about me if John had not pointed me out.

Knuth: Okay. So was it difficult to move, though?

Feigenbaum: No, it wasn't. Well, we had a lovely house in Berkeley, with a grand view of the Bay, so that was tough to leave. But we did. I mean, people do that.

Knuth: Yeah. I mean, you had a toddler. You had a one-year-old daughter, something like that.

Feigenbaum: Yeah.

Knuth: Anyway, you are struggling with other things in life, too, as well as making these big branching decisions. But this turned out to be a decision that was permanent -- you stayed at Stanford and did not keep moving around after that. I mean, this was one of the big forking times in your life, in some sense.

Feigenbaum: Absolutely. And maybe that is just the way I am. Penny and I bought our house in 1975 on the Stanford campus, roughly the same time that you built your house. You built your house a few years earlier, I think, but you have not moved, and I have not moved. We still live in the same places. But what was I going to say about the move? There was a thing which I will call the great attractor, like the astrophysicists talk about. In 1964, Karl Pribram, of the Psychology Department, a neuropsychologist at Stanford, who had been a co-author of a well known psychology book that built on the work of Newell and Simon, Miller, Galanter and Pribram, "Plans and the Structure of Behavior." Karl was running a seminar series, a Saturday seminar series, at the Center for Advanced Study in the Behavior Sciences, which is a think tank for the behavioral sciences on the hill behind Stanford. For those of you who are in the future there, you might not even know it. Karl's seminar was on essentially what you might call AI, or cognition, or models of mind, or just generally that area. And one of the people interested in models of mind that a computer could do was Joshua Lederberg, who eventually became a very close working colleague I worked very closely with. Nobel Prize winner in genetics. He was interested in an aspect of it which was exactly what I had come to by thinking hard about where I wanted to go next in my research, which is to model the thinking processes of scientists, because I was interested in problems of induction, not problems of puzzle solving or theorem proving or anything like that, but inductive hypothesis formation. That is just why Josh got involved with the group of people who were meeting there. He had just come into the field, reintroduced himself to computing via a course he took with George Forsythe on ALGOL, and was interested in that subject. Knew McCarthy, had talked to McCarthy about should AI be on a planet probe that he and other people, Carl Sagan, I think, were planning to land on Mars.

Knuth: Let me try to get this straight. You met these people after you came to Stanford?

Feigenbaum: No. I met Josh in -- it must have been the spring or summer of '64, before I left to go on this big trip.

Knuth: So already, you are coming to Stanford, [and] you knew Gordon Bauer and Pat Suppes?

Feigenbaum: Dick Atkinson.

Knuth: Dick Atkinson, okay, our stellar psychologist. And also Josh Lederberg.

Feigenbaum: Yeah.

Knuth: Even though you are across the Bay, you are coming over here quite often. That is interesting.

Feigenbaum: Tthen when I did come here, the first person I got a hold of was Josh, and we started our project. And Gordon Bauer and I co-advised John Anderson, now a famous psychologist, doing work in information processing psychology area.

Knuth: Okay, great. Now your job at Stanford was quite different also in other ways from at Berkeley, because you were the Director of the Computation Center as well. So this is another aspect. I mean, not just research into studying scientific things, but also a major administrative role, as well. And presumably you would be teaching classes, also, that were different from what you were teaching at Berkeley, so a lot of preparation for classes besides. Is that approximately right?

Feigenbaum: Yeah.

Knuth: So it is a big change in exactly what you are doing everyday.

Feigenbaum: Yeah. Let me give you the timing of it. A little fine grain on the timing. I got to Stanford officially January 1st, 1965. So did Bill Miller, officially, same day. And we were born actually on the Friday after that, at an Academic Council meeting. I was a professor, an associate professor of mathematics for a few days. Imagine that. I want you historians to note that. But anyway, George Forsythe was running both the new department and the Stanford Computation Center, which at that time was going shift machines from a Burroughs B-5500 to something else, I think that was at that time. The "something else" was going to be either a major IBM machine or a set of machines based on a big gift, or it was going to be an upgrade to another Burroughs machine, a faster Burroughs machine. Or it might be a CDC 6400 or CDC 6600. It was going to be a big deal deciding.

Knuth: We had a PDP something-or-other, too, didn't we?

Feigenbaum: There was a PDP-1 that John McCarthy was running for timesharing experiments, shared with Pat Suppes. But it was not available to anyone else except John's group. John was building a timesharing system and Pat was using it for computer-based learning. So George, by mid-year, he just felt he had to give up something. He could not do them all. So he talked it around to all of us, and said, "I have got to give this up. Does anyone want to do this?" Bill Miller did not want to do it. Bill was running the computer group at SLAC [Stanford Linear Accelerator Center].

Knuth: I think George was also president of ACM at that time.

Feigenbaum: Maybe. Maybe that is also the reason. So I said, "Okay. I'll do it." I did it more or less for the same reason that I do a lot of strange things in my life, and maybe for the same reason that I got… Maybe I just liked my experience with Dave Evans, getting sort of immersed in what you might call "real computing" at Berkeley. In this case it would be getting involved with real computing in the sense of buying machines, talking to vendors, discussing timesharing systems in detail, making something happen, blah, blah, blah. All of that stuff. I really got sick of it after a few years.

Knuth: So the salesmen are taking you out to lunch every day.

Feigenbaum: Maybe every week, yeah. But it was very interesting for awhile. Bill Miller helped me out a tremendous amount in making those decisions. That was the time when we did some experiments on SLAC computing to find out what kind of computer was best for their codes.

Knuth: Oh, okay. So SLAC is way up...

Feigenbaum: Yeah. We were trying to unify our purchase, so we were going to…

Knuth: Oh, SLAC had not bought the...

Feigenbaum: No. They had not bought their machine yet. We were trying to decide whether to get a CDC machine or an IBM 360/90 or 91.

Knuth: Oh. In 1965?

Feigenbaum: Yeah.

Knuth: I didn't think the 91… well, maybe it was. I didn't see a 91 until '68 or something like that.

Feigenbaum: Well, maybe they didn't deliver it until then.

Knuth: Yeah.

Feigenbaum: The other one was, Josh Lederberg was promoting a computer for the Medical School. We ended up with a 360/50 there with a so-called mass memory, which was, I think, one megabyte core. I hired Gio Wiederhold, because I knew him from Berkeley, to run that machine.

Knuth: So you were working with Bill Miller, because he was responsible for the computer that the physicists were going to be doing. And on the other hand, the one that you were going to be responsible for was the undergraduates at Stanford, and maybe the administration also?

Feigenbaum: Well, very good point. Stanford was trying to put together... Stanford does everything, or at least at that time, did everything, on a shoestring. People don't realize how little real Stanford money was in all of this. In fact, all of us guys in computer science -- and when I say guys, I mean the women faculty members, as well -- we all had to raise half of our money. I mean, it was only half Stanford money, and half soft money. That went on for quite a long time. Well, the same was true of computing. We had to put together as many bucks as we could find, and one way to do that was to graft administrative computing onto campus computing. Campus computing was paid time from research projects, mostly. A little bit of Stanford money put in there to handle undergraduate and graduate teaching, because the government would not let them get away with faking that underneath the surface. You had to pay for that, just like the government projects had to pay for it, and administrative computing. Then Bill ran his facility, and the Medical School paid for its own facility with grants.

Knuth: And SRI.

Feigenbaum: Oh, SRI also came in. That's right. We made a little deal with them to use up some of our computing time. That's right, I forgot about that. But it turns out the administrative computing was a gigantic headache. It's not all their fault, in the sense that they had other… For them, it was really important that the paychecks got printed on a certain day at a certain time. For us research computing guys, it did not matter if your thing was delayed or… You know, you would feel bad about it, but still nothing bad would happen in your life. If the administrative computing thing did not happen, they were in deep trouble with the government. They were in deep trouble with their employees.

Knuth: I have to say the same thing. I was a consultant to Burroughs, where we were making the machines in southern California. But when Burroughs puts out its own paychecks, they would run a computer that was simulating an old computer that was simulating another one that worked in 1958, because that was the only thing that would reliably make paychecks.

Feigenbaum: So eventually administrative computing and us got a divorce. Then later on they glued them together again, and then they had to get another divorce. Round about 1990 or something like that, there was another.

Knuth: Okay. But you had to make sure that this thing was up 24/7 more or less -- something like that. Now, did you have a choice of making it, like the model that you preferred at Carnegie, where you could push the buttons yourself? Or did you choose a model like you had to use when you were at NPL, where you were submitting…

Feigenbaum: Oh, actually it was not. When I was talking about the British experience, it was not me at NPL that was having that problem. It was the researchers at Cambridge, because Maurice Wilkes was insisting on that mode of operation, where you shoved the cards through the window and got your paper.

Knuth: Oh, I see.

Feigenbaum: But by now, computing was such big business. I have a feeling this machine we bought, the 360/65 -- also a big story there, why it was a 65 and not a 67. I believe it was a 2.5 million dollar machine with an operating budget of maybe 2 million a year. At that point we just could not put that in the hands of individuals. Now, we did later. But I should not say we. I should cross my name off of that, and write in John McCarthy. John, bless him, just kept insisting, insisting, insisting on timesharing for the campus. He eventually got that through the University, to put in money for the Low Overhead Timesharing System [LOTS]. Then dozens of people, hundreds of people around the campus, could have their fingers on the computer.

Knuth: Right. So it was a matter, by that time, being a huge expense in the budget, which was different from when we were undergrads. Very good. Okay. Now I want to mention my plan is [to] sort of fill in the chronological part of this interview, although we are still in 1965. So we have 40 years to go.

Feigenbaum: I will speed up.

Knuth: But I want to just give a reference, so that when we get together again in two weeks, we can talk about the cross cutting things that go through the rest of your life. The way I would say the next major episode is that about this time you said you met Lederberg, and problems of mutual interest. But then by the time I met you in 1968, you had started up a major research lab. So I would like you to talk a little bit about the birth pangs of a heuristic programming project.

Feigenbaum: Don was just saying heuristic programming project. Maybe I should say that slower: heuristic programming project. What was it? It started out being called the DENDRAL project. DENDRAL was the project that Joshua Lederberg and I started in 1965 on hypothesis formation in the area of organic chemistry, based on mass spectrometric and other physical data. We had recruited Carl Djerassi in ’67 or ’66 to collaborate with us -- a famous chemist -- and had good results. Very good results. By 1970, more or less, we (A), had a lot of credibility; (B) we had a lot of confidence, like we really knew what we were doing. We had invented a knowledge representation for the chemical knowledge that we thought was broadly applicable. It was somewhat based on Bob Floyd’s work, and Floyd Evan’s productions, and Newell’s adaptation of that into rule-based systems. So we thought we knew what we were doing. And we thought we needed to play in some other playpens -- not better playpens, but other playpens. Since Josh was a professor in the Medical School, and medicine seemed to be like a good place to look because it was kind of an inexact science. It was not unscientific, but it was inexact. If you are like me, you kind of believe that AI is kind of a qualitative science, not a quantitative science. You are looking for places where heuristics can come into play, and inexact knowledge, and so on. We began to look for medical applications, and found some. Led to a long string of medical-oriented things we did. But at that point, we did not want to be known as... DENDRAL was a project in chemistry. We wanted to be broader, so we needed an umbrella term. The term I invented was “heuristic programming project”, meaning we could cover several of these things. Now, why that? Two reasons. McCarthy was using the word “artificial intelligence” in Stanford Artificial Intelligence Lab, and there was no way I was going to infringe on John’s. He was there first. He is going to use that word. Secondly, boy, artificial intelligence was really a controversial term. It was like touching a nerve in peoples’ body. People were writing books, anti-artificial intelligence books -- Joe Weizenbaum’s book, and there were several others. And DARPA directors would think that it is crazy to work on artificial intelligence.

Knuth: It is scaring people.

Feigenbaum: Yeah. So I just said, look, this is ridiculous. I am just going to call it programming something. There is dynamic programming, there is linear programming, what is wrong with heuristic programming? That is what I do. That is what I called the project. There really were not birth pangs. It was just smooth, what you might say growing a family. The only problem is selling this idea to… Like, how do you sell work on organic chemistry to DARPA? Or how do you sell some other things to NIH that were not medical? It was a balancing act, and Lederberg and I just had to use kind of joint credibility to do this.

Knuth: Were you co-principal investigators?

Feigenbaum: Yeah, on most everything. We would trade off. Like things for the Medical School, like the ACME Computer, the [IBM] 360/50, he was the sole principal investigator of that, and I was the Director of Computing. But we would alternate. On the DARPA project, I was the PI of that, because Licklider knew me. For the NIH project we were do-PIs, and then gave the whole thing over to Carl Djerassi to be PI.

Knuth: Later you had larger and larger staff, and your own computers and things like that. But I am not sure how long it took to…

Feigenbaum: Oh. Well, so then it was in the early '70s that we decided to essentially do what John had done for the Artificial Intelligence Lab. We needed a machine for our work. Josh and I put in proposals, actually a lot of hard work to get NIH money. Then we got Licklider to agree that since we were also an ARPA project -- we were the only non-Defense contract thing on the ARPANET for awhile. We were a medical thing. NIH was supporting it, but Licklider said, yeah, you are one of us. So you can be on the ARPANET. Tom Rindfleisch came from Cal Tech to build and run that facility for Josh and me. It is called SUMEX.

Knuth: Right. So you had research associates who were hired by the project.

Feigenbaum: Yes. Lots of research associates. Very similar to John's situation. That was a big thing at Stanford. You had big laboratories that were operating that way, with research associates. Some of them were research professors. That later fell out of style at Stanford. That later became known as the "SRI style" of doing business. So why are you doing SRI stuff at Stanford? If you are going to do business that way, do it outside of Stanford. It fell into the pattern of faculty-student stuff, and that happened more or less when we moved into the School of Engineering.

Knuth: Are students involved in the early days of HPP [Heurstic Programming Project], too? Or is it mostly the professors doing the research? I mean, do you have...

Feigenbaum: HPP?

Knuth: Like Raj Reddy, for example.

Feigenbaum: He was a graduate student of John McCarthy's when I got here in '65. He was working, using the PDP-1 to do his speech research. I put out a memo, the first DENDRAL memo was joint with a young guy who left us. He was an assistant professor here at Stanford. He left pretty quickly thereafter, named Dick Watson, a systems guy, but he thought he might want to do artificial intelligence. We wrote a joint paper called "Opportunities for Graduate Research." That was the name of it. Came out in the spring of '65, and as a result we attracted two students, two masters students, Georgia Sutherland and Bill White, and they turned out to be major contributors to the DENDRAL project for years. Then we had an extremely hard time -- I mean really hard -- getting any computer science student interested in the real DENDRAL. I mean doing a thesis in the DENDRAL line. the reason was, it involved chemistry. Now, to go back a couple of hours in this interview, I mentioned that when Newell and Simon did GPS, the amount of knowledge they put in was this big: nothing, almost nothing. And here we were putting in just a lot of chemical knowledge, and we had to extract it from the heads of Carl Djerrassi and his people in chemistry. And it was a knowledge… We were inventing knowledge engineering as we were going along. Computer science students felt that they do not do that, that their field is algorithms. Knowledge -- epistemology -- is somebody else's business, like it is a chemist's business, or it is a doctor's business, but it is not their business.

Knuth: Let me try to rephrase it that you could not do DENDRAL with only a small part of chemistry, only knowing a little bit, like carbon atoms or something. You had to know a lot of chemistry. It is not like you could do these trigonometry formulas without knowing a lot of math.

Feigenbaum: Yeah.

Knuth: But you got people like Harold Brown.

Feigenbaum: Yeah. So, Harold was hired.

Knuth: He is a group theorist, but he can study some of the common rhetorical [?] things that go on when you are trying to decide whether two chemical molecules are similar. Or something like this, right?

Feigenbaum: Well, Hal Brown was a mathematician at Ohio State. The ability to attract a tenured mathematician from Ohio State to work on our project was simply sensational. He was just a wonderful addition to our project. A great applied mathematician. We ran into many difficult problems when we moved forward from acyclic graphs to cyclic molecules, especially the rather complicated cyclic molecules that the Djerassi group was working on. Then we had to do a generator for that, and we had to prove that the generator was generating exhaustively and irredundantly. Harold was able to both lead us into how to make the generator, and then going through the proof sequences to prove that it really was doing that.

Knuth: So it has to be quantitative as well qualitative in that sense.

Feigenbaum: Well, our generator had to be. But in chess, for example, the generator does not have to be.

Knuth: No. But the research has... If you just say you are running only on qualitative, I don't know, it seems to me it seems to be a shaky. It doesn't have an actual foundation. There is also some quantitative things which you reach down to. Although it's not your life, it is necessary to have these numbers at the bottom in order to build up the structure.

Feigenbaum: Yeah. Don, I agree with you, but I don't want to agree with you too much, because that is a case that is always over-made in computer science: that in AI, somehow at the bottom there is this real hard nosed quantitative stuff. That attitude is only a very recent one.

Knuth: No, you misunderstand. I am saying that you also need somebody like Harold Brown in order to show quantitatively how this qualitative stuff fits together.

Feigenbaum: Okay. Anyway, the general point is right that we have to hire in a lot of our talent. And that gave the impression, which was okay for a long time, that we are building a research lab with professional staff, and hard to attract.

END OF TAPE 4

Knuth: All right. Now, I am going to try to stick to my goal of moving through the high level instead of indulging all of my instincts to go to the low level details, and move forward. The Heuristic Programming Project thrived, and then changed its name to the Knowledge Systems Laboratory, if I'm not mistaken.

Feigenbaum: Yeah. Let me explain. Very simple. Just as we needed a bigger umbrella term from DENDRAL to HPP, we needed a bigger one because later on in the early '80s because Pitchord [ph?] Liff [ph?] had come back from his residency, was hired as an assistant professor. He wanted to have his own HPP, which he called the Section on Medical Informatics of the Department of Medicine and Research Laboratory. SUMEX had become a pretty big entity on its own by that time, so we needed a bigger umbrella term. So HPP became one of the units of a bigger thing called the Knowledge Systems Laboratory, which was then my lab with Bruce Buchanan, Ted Shortliffe's Lab, and SUMEX. Then we got Tom Rindfleish to run the joint laboratories.

Knuth: Maybe we ought to clarify this for our audience. SUMEX was… well, I knew it as a computer in the Medical School that was a success between another computer called ACME. Bruce Buchanan was a kind of a right hand man for you in heuristic programming.

Feigenbaum: I hired Bruce. Bruce is a very important person in the whole history of this project. I first hired Bruce as a summer employee at RAND while he was doing his thesis in the Philosophy of Science -- a wonderful thesis called "Logics of Scientific Discovery" -- and then convinced him not to take a job as an assistant professor of philosophy somewhere, but to come work for Josh and myself. He eventually became a research professor at Stanford, and then a university professor at the University of Pittsburgh.

Knuth: Okay. So as you say, then these came together and become the Knowledge Systems Laboratory. Approximately when was that?

Feigenbaum: I am guessing approximately 1981, plus or minus one year.

Knuth: After you had moved into Jack's house.

Feigenbaum: I have a feeling it may have [been], yes. After we moved into Jack's. We may have already moved over to our Welch Road location. I'm not sure exactly about the time.

Knuth: That's because you realized that knowledge was somehow the key, more than just heuristics. It is where the talk is going away from databases to knowledge bases. The way I understood it was that databases is a bunch of facts. Knowledge bases is a bunch of meta-facts, or facts about facts.

Feigenbaum: Generalizations.

Knuth: Yeah.

Feigenbaum: Yeah. Actually, I mean, you are quite right. And the date of the… we will get to that. Next time we will get to the "Aha!" moments. But the date of that thing is 1968, when I was writing a paper for one of Donald Michie's symposia in Britain. I felt it was necessary, by '68, to kind of summarize everything we had done from '65. What is the meaning of all those experiments that we did from '65 to '68? Anyway, that turned out to be a thing called the Knowledge Power Hypothesis. Then we kind of tested that out with dozens of projects after that, and came to the conclusion that you don't need much more than an aerosol [?] in the machine. You don't need a whole big logical machine. You need modus ponens, and backward and forward chaining, and not much else -- not much is going on in the way of inference. Knowing a lot is what counts. That leads to things like calling the laboratory "Knowledge System Lab". And even more important, it names things. Like, for example, I tried… In my archives there is a document, that is in this old IBM paper that we were all using at the time on our printer, of a report that I wrote for Steve Lukasik, the director of DARPA, when he was in one of his skeptical modes about AI. Alan Newell asked me if I would sort of… You know, "It's your turn, Ed. Write up the field for Lukasik". And I rewrote the field in terms of knowledge. For example, machine learning was not machine learning. It turned into knowledge acquisition, because I knew that is what the important thing was. It was to the little parameters that an evaluation function was learning. It was acquiring these meta-facts, as you call them.

Knuth: Maybe we should say something about your archives. You have done a great service by recording all these early days in hard copy, and all the things that Lederberg did at that time, and so on. Can you say where that archive report is on?

Feigenbaum: Sure. Let me tell you where mine is, and then I will say something about Josh Lederberg's archives, which are way better than mine. I got the whole idea of saving things for history from Josh. He was really careful about that, and so was Herb Simon. I was pretty careful about it, and I would not only save things, but I would put metadata on the thing so that someone, later on could -- including myself -- could figure out what they were all about. It ended up with a first accession in 1985, roughly, to the Stanford Library Special Collection. Edward Feigenbaum Collection, about 40 boxes of these earlier, acid-free boxes that they used, for which there were audio tape annotations and finder's guides. Then a second accession was done in 2005, and it is still ongoing right now, of another roughly speaking 85 or 90 boxes of things of my career since then, for which the materials are still being prepared. Slow process. Expensive, too. It is going to end up in digitized form. The library is digitizing a lot of the material right now, and it is going to be a website similar to the Herbert Simon, Alan Newell website. However, there is one very special website that people ought to… Well, you people 50 years from now, maybe it will be gone. But the National Library of Medicine created a series of special autobiographical websites and archival websites, of which the first one was Joshua Lederberg's website. If you go to Google right now and type in "Joshua Lederberg", the first hit will be this nlm.gov Profiles of Science [profiles.nlm.nih.gov/bb/] . There is a absolutely beautiful one. Josh did one for himself. He did one for Oswald Avery, the discoverer of the DNA's genetic material. And then NLM has done several more, like a dozen more, for Nobel Prize winners in biology and medicine.

Knuth: I met somebody a week or two ago that said he was working with you on putting this into the semantic web.

Feigenbaum: Yeah. I am co-directing a project along with Mike Keller, the Director of the Stanford Libraries, to make a software system available that will allow not just special people like Lederberg, but anybody to organize a professional archive and make it available in an accessible form. Maybe be widely distributed by, if you are in computer science, maybe by the Computer History Museum, or the ACM, if it is some other field. There is a model of it, which is your software on digital typography, which is now used by many fields, not just mathematics, for typesetting. The idea would be to do many fields here, except here it is a little… There is a wiggle on it, which is that it is one thing to build a conceptual structure that fits other computer scientists' lives. They do not even fit exactly. It fits more or less exactly with John McCarthy's life -- my life and John McCarthy's, more or less -- not quite so much with yours. But the gap to a biologist or a lawyer or a historian would be a lot more. So there is going to be some ontology work needed to be done for these other fields.

Knuth: That sounds like a great service to all scientists. Moving right along. You were chair of our department from 1976 to 1981, if I am not mistaken. That was a pivotal time in the history of Stanford computer science, because we moved into a new building in the center of campus from many sites all around the campus. Maybe you can say just a few words about how you viewed that particular part, as far as serving the department is concerned, as opposed to working in your lab.

Feigenbaum: It is a tough thing to talk about. I rarely do things I do not like to do. So I have got to say, historically speaking, it must have been the case that I really wanted to do that. It must be because of either the legacy of Forsythe, or maybe it was my feeling that the department needed something in addition to what Bob Floyd had done. Or maybe, I don't know what. In my professional life, it was an enormously productive time. My second wife Penny and I had gotten married in '75. That was like kick-starting a new life, and it was wonderful. And it kick started everything else. Things were really booming along. That is when I started the Handbook of Artificial Intelligence project that resulted in four volumes later on, published in '81 through '84. And DENDRAL, the second half of that. The DENDRAL work was coasting along under Carl Djerassi's PI-ship. Not only had we done a lot of work on MYCIN, which was quite well known, but by that time we had so much confidence that we were trying everything. I did a sonar signal processing application. It was a very, very difficult one. It was done for DARPA in a classified way, but I had to figure out to get those new systems described in a way that would not violate classification. They were excellent systems. We did extra crystallography. We did other engineering applications. We were distributing software like crazy. Then we started some companies to take that on. It was an extremely productive time.

Knuth: In spite of the fact that you are also chair of the department.

Feigenbaum: Yeah. And I just cannot imagine (A) why I did it, and (B) how I did it. I know that I survived because I hired Denny Brown, the wonderful, young student actually. He did not finish his PhD, but he was just fabulous in knowing our department and helping me through. We just decided that the one thing the faculty did not want was committee work. So Denny and I would do the whole thing. Except for admitting new students, we would just do all the committee work and would not call. You remember that? I think you came into thank me once for that.

Knuth: I am sure I did.

Feigenbaum: I just don't see how we did it. It was a very stressful time, because the one thing Bob Floyd and the rest of us had agreed to do was to move into a space which was too small. We were doing it on the basis of a theory which we had learned from Don Knuth, which was called the "sardine can theory": that you are most productive when you are most stressed in a sardine can. Maybe that was it. But it was really stressful in that sardine can.

Knuth: It was actually because the field refused to stop growing.

Feigenbaum: We had gotten so well known in our area that money kept pouring in. Licklider was long since gone. He had told me at one point to double my budget. He said, "Why are you asking for so little money?" Later on Bob Kahn basically did the same thing. So we were growing like crazy. We had to move out of Margaret Jacks Hall into another place, a rented space, a location near Stanford. But it was very stressful. So during that period, I actually took two one-quarter sabbaticals. The reason being. I was just too stressed out. I could not stand it. I had to go. One time I went off to Burr [ph?], and the other time to Paris for a quarter as a consultant for Schlumberger doing an expert systems there. Finally I just gave it up. I really had two three-year terms. I had a three-year term and it was renewed for three years, but in '81 I just said I cannot do this anymore.

Knuth: Was that your first sabbatical?

Feigenbaum: I don't remember.

Knuth: A sabbatical usually is every seven years, but you started in 1969.

Feigenbaum: Yeah. I may have taken... Almost all my sabbaticals I took as a leave at Stanford, living here without moving.

Knuth: I see. So, I guess we're understanding the timeline pretty much. We're getting into the 1980s, when all of a sudden a lot of companies are being started to make available all of these successful kinds of software known as Expert Systems. We will talk next time about Expert Systems in detail, of course, because this was another big world-changing thing. But zooming between then and now, I'm thinking that also there were other times when you took leaves of absence, especially for national service. Can you sort of just give a timeline from 1980 until now, as how your life divided?

Feigenbaum: I would put them in these categories. I would say that the… it is hard to find words for it, but the numbers are roughly speaking '83 to '90 -- or '84 to '90 -- was a period in which I made a decision to change course in the kind of research we were doing at the Knowledge Systems Lab. I mean personally, not the lab. Personally I decided to change course into parallel computing, in particular multiprocessor parallelism, not massive parallelism thinking machine, but more like what Gordon Bell was doing with five processors, 20 processors, something like that. I wanted to see to what extent artificial intelligence techniques, software techniques, could be adapted to that kind of a parallel world. Especially the techniques called "blackboard techniques" which were so powerful. It is the most powerful thing we have ever developed and worked on. It turned out… I mean there is a whole story behind that, how successful we were moderately, and why we were not more successful and why are we so successful now. It turns out the world has changed from a massively parallel view of parallelism to four cores on a chip. That is the kind of thing we were working on. In a way what we did then is more relevant today than it was to the '90s. Then came a stretch of time when I was disassociating myself from the day-to-day running of the laboratory. Because I had made a decision around '89 -- it was pretty much the same decision Marvin Minsky and John McCarthy had made a lot earlier -- which is they were not going to be responsible for the fundraising for the laboratory anymore. It is too much work.

Knuth: It sounds like what I told you I was doing in 1989.

Feigenbaum: Yeah. Les Ernest for a long time was doing that for John. Raj Reddy was doing it for Al Newell, and so on. Bruce Buchanan and Bob Inglemore [ph?] and Rich Fikes, later on, began to do it at the KSL. That was from like '90 through '93, early '94. But in '94, an amazing thing happened. I mean a lightening bolt. The phone rings and it is a person that I knew, but I had only met once, from the American Academy of Arts and Sciences -- one of the members, Sheila Widnall, a faculty member at MIT. She calls up and says, did I know anyone who wanted to be chief scientist at the Air Force? And by the way, if you are interested, let me know. She did not tell me why, but it turned out that she had been already picked by the Clinton Administration for being Secretary of the Air Force, and she was looking for her chief scientist. I talked to my wife about it overnight and called her back the next day and said, "Yes, I am going to do that." She wanted a commitment of three years. That is more than normal. Normally they like a commitment of two years, but she wanted three. And I said, "That is okay with me. Yes, I will plan on that and we will take a look at it at the end of two." I actually stayed three years doing it.

Then I came back to Stanford. It was a rather unsettling experience to come back to Stanford. Partly you come back Stanford after playing out on a big stage, and then all of a sudden you come back and they look around and say, “What are going to teach next year?” As opposed to, let’s say, Sheila. When she stopped being Secretary of the Air Force, she was given a university professorship at MIT to basically do whatever she wants. I had to go back to teaching the normal kind of AI stuff, and that is a lot of work actually in a field that is rapidly changing as ours. The other thing is Rich Fikes was doing a great job running the lab, and he had his own people, and I was not about to start up anything new at that point. So after a relatively short time, roughly speaking the beginning of 2000, end of ’99, or somewhere in there, I retired. Since then I have been leading a wonderful life doing what I damn please, and doing this project with the Library, and doing a lot of consulting for the Air Force.

Knuth: The Air Force was also the sponsor of RAND. Did you have connection with Air Force all the way through, or was this sort of a new thing in 1999?

Feigenbaum: The answer is yes and no and yes. I had been involved with Air-Force-like problems, by being involved with RAND as a consultant over the years. On the other hand, periodically and routinely the Air Force had asked me if I would be on the Air Force Scientific Advisory Board, SAB, and routinely I would turn that down, that being a job which is very time consuming -- that is, time consuming in the sense of about 20 to 25 days a year. There was no particular time in my life when I had 20 to 25 days a year to devote to that purpose. So I kept turning that down. Bob Cannon from the Aero-Astro Department was a friend of mine -- has been a friend of mine for a long time. He was the one who first told me what a wonderful time he had had in 1965 coming out of a deanship at Cal Tech. He went to become Chief Scientist of the Air Force. I said, “What’s that?” This was a long time ago. He just told me about that. So when Sheila called I was all set to think positively about that job. Bob was right. Sheila was right. I talked to Bill Perry -- he was Defense Secretary. He may have been Deputy Defense Secretary at that moment of time, but he became Defense Secretary. I talked to Bill Perry and he said, “Ed, what better time?” Anita Jones is CDR&A. I am DOD head. This is a great time to do it. So it was.

Knuth: So, are you spending most of your time at the Pentagon, or are they flying you around all over the world talking to troops?

Feigenbaum: Yeah.

Knuth: Or who were you talking to as chief scientist?

Feigenbaum: The Office of the Chief Scientist is in the Pentagon. It is in the Air Force portion of the Pentagon in a big-shot designation place. That is, it is on the outer ring where you get windows that look out on the real world. But the job actually, when I first went in to talk to my boss, the way the hierarchy works, my boss was the Chief of Staff of the Air Force. It was not Sheila, actually, even though she hired me. It was the Chief of Staff of the Air Force. When I went to talk to him, one of the first things he said to me was, “I want you out of this building. You don’t learn anything in this building.” I took him seriously, and just went everywhere. That took me to a large number of Air Force bases, and it took me to other countries doing liaison with, let’s say, the Israeli Air Force and their scientific staffs, or helping the DOD build relationships with some members who were going to join NATO, like Poland, Hungary and the Czech Republic. I went to all of those. I gave a talk for the Air Force, a kind of keynote speech, at a conference they were running in Saint Petersburg. Many, many other things like that. So I was away. In fact, one of the reasons I did not want to get a really expensive apartment in Washington was I knew I was not going to be spending much time in it, because I would be going to places like Cheyenne Mountain.

Knuth: What is Cheyenne Mountain?

Feigenbaum: That is where the Air Defense Command is. But that is also the place where they had a gigantic software snafu. So the question is, why did they have that software snafu and what happened there?

Knuth: I see. So you also do some debugging work.

Feigenbaum: That was more like organizational and contractual debugging.

Knuth: As you go to these other places, it is so that you can bring back knowledge from there to the Pentagon, too?

Feigenbaum: Yeah. I was a member of the Air Staff, which means that I was invited to all the meetings -- actually daily, more like maybe three times a week -- of the Chief of Staff and the four-stars and three-stars that were running the different big blocks of the Air Force. I would sit in on the meetings to the extent that I was in the Pentagon those days. Then I would be writing memos to the Chief, or going in to talk to Sheila about something, or going downstairs to talk to Anita Jones about something that was a DOD wide problem.

Knuth: What level of security clearances?

Feigenbaum: Top Secret. I did not have any special clearances that I remember, except once, when I had to get something really special to go into see the Atomic Bomb History Museum at Los Alamos. That was awesome. That was just awesome.

Knuth: But you needed special clearance to see that, you said?

Feigenbaum: Actually, to tell you the truth, I really never got that special clearance. They applied for it. It did not come in time, and they just sort of broke the rules to take me through that door.

Knuth: Well, we will never tell anybody.

Feigenbaum: No, you will never tell anyone in history. But, anyway, it was a fantastic time. Not only that, but also asking the Air Force to fly me in an F-15 and an F-16. I actually flew a B-52 bomber. I was on a war game, and after the virtual bombing was done, I flew the B-52 back to its airbase and did virtual bombing myself.

Knuth: I am just trying to check this out. You were 60 years old or something! This scares me.

Feigenbaum: Anyway, great, great time at taxpayer expense.

Knuth: So you can look back on that time as something that you were able to give back to your country, or something like…?

Feigenbaum: First of all, I felt that way. And second of all, it was a buoyant time to be there. I think the Clinton administration was a buoyant time, until the Monica Lewinsky thing and the impeachment thing came up. But it was a pretty buoyant time in Washington. Everyone was feeling good about everything. The war was over. Cold War was over. Things were changing. Bill Perry -- wonderful, wonderful person and marvelous Defense Secretary -- was running the show. An MIT professor was head of the Air Force. I mean, what more can you [ask for]? It could not be better.

Knuth: Could you also make strategic..

Feigenbaum: Yeah. That was the other thing I was going…

Knuth: … improvements in the Pentagon computers?

Feigenbaum: That was the key thing that felt really good. I was the first person to be asked to be Chief Scientist who was not an aero-astro person, or a weapons person, or a laser person. There had not been any computer scientists before. What Sheila had in mind was she wanted a software expert in there, because the Air Force… She looked at the Air Force as having a giant software problem. When I took a look at it, it was giant software problem squared. What I was able to do there... I really did two big things, I think. Tthree big things. One was consciousness-raising about software. In fact, the one big report I wrote at the end of my term, the one that got circulated and got briefed to all these generals, was a report called, "It's a Software-First World." The Air Force had not realized that. They do not think that. They think it is a titanium-frame-based world. They do not understand that those things just fly because of the computers in them. That is one thing.

I said three, and I am going to probably only remember two. But that was one of them. The other one was software development. They really have believed, and in fact, the contracting world that they structure for themselves and deal with, can only imagine a structured programming top down world. They cannot imagine… Because they know how to do that. You set up requirements, and you get another contractor to break up the requirements in a block, and another contractor breaks them up into mini blocks, and down at the bottom there are some people writing the code. It takes eight years to do that code. When it all comes back up to the top, (A) it does not interact with… it does not fit. And (B) it is not what you want any more. They just don't know how to do the contracting for cyclical development. No matter if Barry Boehm could be talking. Barry Boehm is a software engineering expert at USC. He could be talking to them forever about cyclical development methods, and they won't know how to implement it in their contracting structure. Well, I think we were able to work that through, and got the main Air Force procurement place, that is in Massachusetts near Boston, to get that all figured out how to do that. That was really good. What else about software? I never did really figure out to get AI to… I mean, not that the Air Force didn't have it, or the Army didn't have it. The Army actually had an AI center in the Pentagon. But it never became a big deal.

Knuth: This was before there was a lot of things about sensor networks and so on.

Feigenbaum: No, that was another big deal. Yeah. Sensor networks, and integrating what is coming in through the sensor networks with what I would call, as an AI person, centralized model building. They would call it decision aids for Command and Control.

Knuth: Centralized instead of distributed.

Feigenbaum: It was not distributed, at that time, anyway. I don't know how it is now. But another thing that we did that I thought was very, very useful… It was Sheila's idea. I only contributed one nth to it. Sheila decided that it was, I think she said, 50 years since von Karman had written the original blueprint for Hap Arnold as to how science can aid the Air Force, science and technology can aid the Air Force. Von Karman was, he was not called chief scientist, but he was like the original one. He was called something else. So he was chief scientist No. 0. And 50 years later, Sheila said it is time for a new plan. Von Karman's plan kind of ran out of gas in '62. They went through the whole plan, which was a brilliant plan, never mentioned the word "computer", 1946. So Sheila put together a study run by the Scientific Advisory Board of all the areas of science and technology of impact to the Air Force. I ran the one on information, on the computer science part. Not the information applications, which was a separate volume. So I had people like Bob Lucky on my committee, Paul Saffo, Gio Wiederhold, a variety of very good people. That was a very useful study. It became the framework for the Air Force spending money in that area for years.

Knuth: Okay. Well, thank you very much, Ed. If people watching this are anything like me, we will not be able to wait two weeks, like I will have to wait, to see the next episode. We've got all the dates filled in now and the framework, and next time we are going to go in and put lots of threads into this warp and waft here that we have got. So thanks very much for this first half.

Feigenbaum: Great.

END OF TAPE 5; END OF INTERVIEW 1

Interview 2: May 2, 2007

Don Knuth:I want to welcome everyone who's watching or listening or reading the second part of this oral history with Ed Feigenbaum [on May 2, 2007]. In the first part, we went pretty much chronologically starting with when you were born and ending with the national service that you did after your retirement from Stanford. Now today what I want to go through is sort of cross-cutting topics, certain themes that happened throughout those years. It's easier to organize some parts of your life chronologically, and other parts by topic. So it's topics today. The first thing that strikes me, if you had to summarize, a lot of your career was based on the idea of heuristics. Studying heuristics is a great metaphor for doing artificial intelligence research. But what I want to ask you is: what were your own heuristics for the personal choices that you made through your life? The way it gets through to you, instead of the scientific, but also personal?

Edward Feigenbaum: Don, that's a really good question. There was a time in my career when I would try to write them down, and give students lectures on that kind of career direction. And those will be -- not some, they will all -- be in the Stanford Archives, in the special collection of my stuff. So right now I'm just going to try to spin off a few off the top of my head. Where shall I start? I think I'll start with one that I've talked to you about earlier, which was that I found myself to be a better discoverer than inventor. Which meant I needed to pay a lot of attention to empirical data, because in empirical data, one can discover regularities about the world. Which means that my methodology kind of forced me to pay attention, really carefully, to the real world. So not to move off in fantasyland in science, but to stick pretty close to reality. That was one thing. The other thing was never move off into a complicated problem area unless you have a problem domain that you can get data from, that you can work very closely with. I saw it at work in my graduate training under Simon, and working with Newell on their chess playing program. Even when they were doing GPS, they still were using concrete task domains like trigonometric identities. Then I spent years looking for the right task domain for studying the scientific induction problems that later became the Dendral Program. That was a key thing throughout my life, to look very carefully at the problem areas. Some would be terrible. If you wanted to work on a certain kind of problems, some problem domains would be terrible. Some would be extremely good. Newell, who was always impressed with the Dendral Project -- but not so much with the brilliance of the ideas in the Dendral Project, which he thought were interesting and a good coupling together of other ideas that had been in the field -- but with the choice of task domain, and the way of exploiting the task domain to get the results. So, that's an important heuristic…

Knuth: Is it easy to give an example of a bad one, of a bad domain?

Feigenbaum: I'll give you one that I really rejected when I was searching for the domain that ended up being mass spectrometry because of Lederberg's suggestion. A domain that I was looking at for studying induction was to induce the rules of baseball from a long string of specific activities in baseball. It turned out that that just didn't feel right. One of the things that made it not feel right -- and there comes another heuristic -- is that I've always wanted to, in doing this, work with people who are genuine experts in that domain. In the baseball thing, it didn't fit. What I was trying to do was induce the rules of baseball.

Knuth: You didn't know Dutch Ferring. He was a friend of mine. But you knew Josh Lederberg. Dutch was a Stanford legendary baseball coach.

Feigenbaum: No. No, I didn't. Another heuristic that I used was -- how to put this-- I really liked to meddle in other people's domains of work. I just find it absolutely fascinating. Which means that there are large chunks of computer science itself which I am not interested in exploring, relative to, let's say, exploring physics, like you and I were talking about, or molecular biology or chemistry or some other domain. I got involved in many, many domains -- x-ray crystallography, engineering domains. I find it fascinating to open the door into those domains, become a rapid expert -- not a real expert, just sort of a 30-day wonder expert in the domain -- and find out about a lot of things.

Knuth: Is it that you liked the idea of finding a pattern in something complicated, taking something that's organic and messy and bring some order into it?

Feigenbaum: Yeah. I think that's part of it.

Knuth: If it's already clean, you might not feel attracted to it because you're better off finding out how to deal with the more… I mean, baseball itself is a good example of something that is complicated -- lots of nuances to it -- but it doesn't have a structure where you wanted to do something about it.

Feigenbaum: I'm not sure that I thought it through that carefully, but I think you're right.

Knuth: So this is like an experimental physicist [who] has a complicated universe to observe and wants to try to see the patterns in that. Is that right?

Feigenbaum: I think I'm much more like an experimental physicist than I am a theoretical physicist. A lot of my computer science colleagues, and indeed our students who come from mathematical backgrounds, are much more like theoretical physicists. In fact, I kind of named a disease that they have, which has gotten popularity because Doug Lenat has used the phrase a lot. It's called "physics envy".

Knuth: Physics envy.

Feigenbaum: Physics envy. Our friends in computer science have a lot of physics envy, and heuristic programming is kind of messy. That's, in fact, what Polya was trying to tell people back with his books on "Patterns of Plausible Inference" and "How to Solve It" -- talking to his mathematical colleagues and saying when you prove a theorem and you get the QED, it's not all that simple. You've done something a lot more messy than that to get there. You've searched a big space, you've discarded lots of avenues in the space, you've used a lot of rules of good guessing to get there. But you just don't want to tell people about the messiness of that. So the whole idea of using heuristic methods was to employ all those armaments, you might say, in the service of pruning these enormous spaces.

Knuth: Computer science has its characteristic of non-uniformity. I mean, a computer program has step 1, step 2, step 3, step 4. Well, a lot of the pure mathematical things, they have one equation that applies to everything. To a mathematician, the hardest thing is you have a theorem that you have to prove by considering ten different cases. But in the heuristic kind of work that you're doing, you've got thousands of cases and still, you're interested in finding some paradigms that enable you to deal with these because they don't just go away. Not everything in the world is uniform.

Feigenbaum: Let me give you another example of a heuristic that I was using. I used it some times, and I didn't use it other times. When I used it, it was successful, and when I didn't, I wasn't so successful. And that is: it takes a while to become really, really good at something. Stick with it. Focus. Persistence on problems -- not just problems, but a whole research track -- is really worth it. Switching right in the middle, flitting around from problem to problem, isn't such a great idea. The thousands of little intuitions and pieces of heuristics you gather along the way for dealing with a particular problem area, or your part of computer science, are enormously valuable in making the next step and choosing next steps that don't waste your time and your students' energy and your own energy. Of course, I violated that by doing a lot of miscellaneous things in my career, which de-focused from those things, but I enjoyed them and did it anyway.

Knuth: There are some people who are really good at inventing a theory where it's needed -- I'm thinking of Jeff Ullman, for example -- and then going to anther topic a year later and seeing where another area needs a theory, and so on. But those are very specialized. You started with the work on mass spectrograms and then went into medicine. I've always wanted to know if you ever thought of, like, bubble chamber and cloud chamber [images], the things which are somewhat analogous where you have a lot of data.

Feigenbaum: The answer is yes, we did. Actually didn't work on that particular problem, but worked on similar ones in medicine. But the reason the one you mentioned that was salient was that SLAC [Stanford Linear Accelerator Center] was being built or had just been built when I got here. Bill Miller was running the SLAC graphics group, and so we had quite a bit of conversation about that. It would have been actually a very good problem area. I'm not exactly sure why we didn't move down in that direction. It's just that I had…

Knuth: But you never claimed them as Artificial Intelligence, as if they were also working on AI

Feigenbaum: See, I think that the problem -- I'm just guessing now -- but the problem may have been that the physics of the dispersal of the particles in the bubble chamber or cloud chamber was too well known.

Knuth: It didn't need more funding, or more people working on it.

Feigenbaum: The AI field, as Nilsson and others have said over the years -- and Roger Shank -- kind of breaks down between the neats and the scruffies.

Knuth: The what?

Feigenbaum: The neats and the scruffies.

Knuth: Neat and scruffy, okay.

Feigenbaum: McCarthy represents the neats and over the years. Newell and Simon, and Minsky, and myself, and Shank, and others represented the scruffies, where we don't insist on a lot of formal clarity in the models. Well, in the case of the bubble and spark chamber photos, there was just a lot formal clarity there, period. You didn't really need it. But in medicine there wasn't, so it was kind of a good area.

Knuth: Yeah, okay. But there was more of a need for it, in that you could see there were resilient physicists working on…

Feigenbaum: But I think that what I just said overstates the case in the following way -- and again, it's a heuristic in life. Meet a wonderful collaborator -- and in this case, it was Joshua Lederberg -- meet a wonderful collaborator, and really work with that collaborator on some meaningful problems. Lederberg and I worked on a whole chain of problems. Then even when it wasn't directly in our field, we were involved with that, like in diagnostic medicine, or Josh on the engineering side of what we were doing. That's a valuable heuristic. Another way to put it is that, as we discussed in your oral history, you said you were pretty much of a loner, that you liked to work -- you gave some examples of where you weren't, but you said most of the time you liked to work by yourself. One of my heuristics is liking to work with a small number of very smart other people, like Carl Djerassi or Josh Lederberg.

Knuth: That actually was a good segue to the next thing I wanted to ask you, was different gurus. But I guess the word "guru" isn't right. You mentioned Herb Simon many times last time, and he was definitely a guru for you. As an advisor, I found I couldn't work with my own PhD advisor on a same level. But with Lederberg, you were more of a team. Would you describe other gurus in your life or other…

Feigenbaum: Sure. Great question. In fact, myself -- in trying to structure my past in a reflective way, perhaps in that [way] which might lead to an autobiography -- I've thought of structuring it in terms of people I've worked with, and people who have influenced me, and people I have influenced. No question that the two biggest influences on the way, I think, were Herb Simon and Joshua Lederberg. Simon's influence was absolutely fundamental in two things: getting me interested in the breadth of the world's problems, and not being too concerned with the formal details of the analysis of it. That's why I said earlier, I think in my last session in the interview, that I thought of AI more as a behavioral science than as a mathematical science. That kind of came from Herb's breadth.

Knuth: With Josh, did you go to him, or did he come to you?

Feigenbaum: No, I'll tell you exactly how it happened. If I had a copy here of the book *Computers and Thought* I would read you a paragraph at the end of the introduction to *Computers and Thought*, which gives you my projection, or my statement, about what I thought was the fundamental importance of the problem of induction. I was looking for a problem domain with which to work. It wasn't an intense problem because I happened to be, at that time -- for reasons that are historical -- I happened to be teaching in a school of business administration, where I didn't have any great graduate students anyway, so it wasn't a pressing problem. So it took a lot of thought. In the middle of 1964 I was invited to a series of Saturday morning seminars -- workshops -- at the Center for Advanced Study in the Behavioral

Sciences, which is on the hill behind Stanford. Carl Pribram was running that. Carl was a neuropsychologist who had participated in writing an early book on information processing psychology -- what we now would call cognitive science. The book was called *Plans and the Structure of Behavior*, [by] Miller, Galanter, and Pribram. So Carl was really involved in that field. He was running this workshop on what amounts to models of minds -- whether they were artificial minds or natural minds. And Josh had gotten interested -- I should say re-interested -- in computing. He had been interested in computing at an early stage of his great career, but then he put it aside because he had other things to do in the fabulous work he did that won him the Nobel Prize. But he got back to it in, roughly, 1964; taking a course from George Forsythe in ALGOL, and using a problem related to organic chemistry as his test domain for testing out his understanding of these ALGOL ideas. Josh knew McCarthy. He was interested in the subject, and he showed up at this Saturday meeting, and that's when I met Josh. As soon as I talked to him about my interests, he immediately had suggestions to make. One of them was concerning mass spectrometry. He was working on it in relation to putting a planet probe on Mars, looking for signs of life, precursor molecules. The idea of just beaming the data back to earth for humans to analyze didn't appeal to him. Put the software on board and let the planet probe tell you what the structures of the molecules were. That was in '64. When I actually showed up at Stanford, aside from saying hello to Forsythe and McCarthy, the first person I looked up was Lederberg, and the collaboration began almost immediately. He remembered our conversation. I remembered our conversation. It seemed like the right thing to do. We just fell into it.

Knuth: This was before he had his Nobel Prize?

Feigenbaum: No. He won the Nobel Prize in 1959, I believe it was. Then he gave all of his Nobel Prize speeches and all that. Finally got sick of traveling and decided to look for another field to work in, and he got interested in computing and artificial intelligence.

Knuth: But when he won the Nobel Prize, he was in his 20s or 30s?

Feigenbaum: Very young. Very young person.

Knuth: You told me a story once about his stack.

Feigenbaum: Oh yeah. Sure, I'll repeat that. Josh: I know if you're listening, this is my interpretation. You don't have to buy this at all. I just saw Lederberg yesterday in New York, and he said he was interested in hearing this oral history. One of the questions that I and others had been looking at during the '60s and a little bit later in the '70s… Herb Simon was the leader of this; I was more of an AI engineer, you might say, and he as more of a psychologist. But the question was: what is it that distinguishes experts from novices? Herb did some experiments, and came up with some ideas. I was asking the same question of myself when I would watch Josh Lederberg at work -- sorry, when I watched his mind at work, or listened to his mind at work. What is it that makes Josh so amazing, so penetrating, so deep, looking into problems? And knowing what I knew about problem solving, I made a conjecture. Which is, that the portion of Josh's immediate memory, which serves as a place to return to sub-problems, was just a lot bigger and more efficient than the rest of us. He could keep a deeper stack. Maybe I should say it another way for the people watching this oral history. One of the key techniques that AI people -- the models that AI people use -- in problem solving, is what's called sub-problem reduction. Divide and conquer. You don't solve a big problem all at once. You solve it by dividing it up into sub-problems, which you divide into sub-problems. You have to keep track of that whole chain of what we call the sub-problem hierarchy. And Josh just had an amazing way of keeping track of where he was in a deep line of thought. My feeling is that can be as genetic as any other thing. I don't think of myself as being able to run a hundred-yard dash fast, or a four-minute mile, or anything like that, because I don't have the musculature to do that. There are some things that are just genetic and maybe Josh's deep, deep stack is something that's genetic. Or maybe he figured out how to train himself to do that when he was young.

Knuth: It came up in the other interview too, when we were talking about levels of abstraction. It's very similar, [in] that you have to know the state of lots of different things at different levels.

Feigenbaum: Don, another thing that comes up -- you mentioned it in your interviews -- is the level of concentration you need to do that. You can't be interspersing other symbols with that stack when you're doing it. It's too hard to do both at the same time. I guess the lesson for kids out there watching this interview is: don't listen to a lot of music while you're doing complicated things, because those are just other symbols which are coming in there and bothering your stack.

Knuth: You can listen to baseball games though. Sometimes it helps. I'm just kidding, since baseball came up. When you're doing something boring, some kinds of distractions do help. But when you're doing science, the interruptions are killers.

Feigenbaum: Killers, right. Absolutely. So therefore you find people who take deanships. Like one of our former deans at Stanford said, I remember, when he took the deanship, "Yeah, that's okay. I can do that part time, and the rest [of the time] I'll run my laboratory." No, there's no way you can run your laboratory when you're doing that. You just lose it.

Knuth: Any other gurus?

Feigenbaum: Djerassi was another one. The way Josh and I got Djerassi was involved. Let me tell you a little bit about Carl Djerassi. He's a brilliant person. I once heard Lederberg call him "a phenomenon of nature".

Knuth: I was once having lunch with Josh, and I remember exclaiming, "Josh, how could you have ever evolved?"

Feigenbaum: Right. Remind me to come back to that question -- that exact thing you just said -- because I want to make a comment about that. That's another heuristic, actually, of my life and career. But anyway, Djerassi was the so-called "father of the birth-control pill". He then went on to run Syntex Corporation. Actually, his whole career at Stanford was spent 50% being a professor, 50% running a company like Syntex or one of his later companies at Stanford. He was one of the world's great collectors of Paul Klee paintings, and just a very broad intellect. But when we got to know him, he was a mass spectometrist. One of the best. He had a laboratory over in the chemistry building. Josh said, by 1996 somewhere, like one year into our project, Josh said, "Look, Ed, we're running out of what I know about mass spectrometry. In order to make any progress we're going to have to get someone who really knows this. Let's get Carl." We had a little strategy, which worked. We took Carl up to the Stanford AI lab where our program was running under timesharing on McCarthy's machine. We showed him the program working on… First, we showed him the good stuff, which was working on some amino acid spectra. Then we showed him the program running on alcohol spectra, and it was doing a terrible job, even though alcohols are much simpler. But it didn't know anything about alcohol mass spectrometry, so it was doing a terrible job. Carl said, "That's terrible." And we said, "Yeah, that's right, Carl. Now, what is it that you know about alcohols that our program doesn't know?" That got Carl hooked. From then on in, it was just Carl and his team supplying that kind of knowledge. Then we had a program of research that Carl laid out that extended the range of Dendral's abilities. Each time that we went into something more complicated, we had to do more complicated stuff in the program. Like at one point, we went into ring structures, which means that we had to get Harold Brown in. He came from Ohio State to help. And some other people, Larry Masinter, to help figure out the… And we consulted with you, and we consulted with Bob Tarjan about understanding the graph theory of the ring structures and getting that into a program. Carl was a big influence on what we did.

Then I would say another influence was a person I influenced a great deal, but he also influenced me a great deal, and that was Bruce Buchanan. I found Bruce luckily. Stroke of sheer luck. Bruce was a graduate student in Philosophy of Science at Michigan State. He applied for a summer job at Rand. I was working in the summers at Rand at that point. Paul Armer, who was hiring people, gave me this application and said, "Ed, would you like this person for the summer?" It was just when I was thinking about scientific induction, the reason being that Josh's comments had turned me in that direction. It was just a natural to use scientists to study induction because they were professional inducers. That was their job, to look at a lot of data and construct hypothesis and theories. So Bruce came along with this thesis he was writing called *Logics of Scientific Discovery*. Listen to the words: logics of scientific discovery. That was exactly -- I mean, it couldn't have been more tailor-made. Bruce came in the summer. Then when he wanted to get a job as an assistant professor of philosophy somewhere, he asked me to write a letter for him. I just said, "Bruce, you don't want to do that. What you want to do is come and work with us. " I told him what the project was, and he did. And over the years, it was a really great collaboration. Bruce had just the right flavor of insight from his background in philosophy and his work in computer science to contribute to our work. So that was excellent.

Knuth: I don't know how to ask this so you can answer modestly. But other people who would put <u>you</u> on their list of gurus.

Feigenbaum: Yes. The answer and a lot of them showed up at the 70th birthday festschrift. It wasn't kind of a schrift; it was just fest. Doug Lenat is one of them. His graduate student career was kind of languishing under Cordell Green. It wasn't getting anywhere. He somehow discovered me and described what he wanted to do. It fit exactly what we wanted to do. He just went straight ahead. That was his AM thesis, and we've had a very good collaboration over the years. Another one is Randy Davis. Randy's work on the explanation. I don't know who we would… But there are many.

Knuth: I'm thinking Peter Friedlan [ph?] maybe.

Feigenbaum: Oh, Peter as well, sure. Sure. Peter and Mark Steffic [ph?] did wonderful theses. Well, there's a whole story there about the so-called MOLGEN Project, molecular genetics. It was the first project in which computer science methods were applied to what is now called computational molecular biology. The reason we were able to get in there early is that we had Joshua Lederberg as our collaborator. The genetic engineering methods had just been invented by Stan Cohen and Herb Boyer. Stan Cohen was one of the co-PIs of one of the projects of the Meissen project. So we were there very early. We were masters of a computer language that was way better at handling symbols than it was at handling numbers, named Lisp. But that's what their domain was. It was all symbols. So we were able to give them tools, [and] they were able to give us insights. Peter and Mark did the first two theses on using AI methods for experiment planning in molecular biology.

Knuth: You had the vocabulary and the kind of theory that was needed to talk about this kind of modeling. Doug Brutlag -- was he involved?

Feigenbaum: Oh, Doug Brutlag. So, here we were working in molecular genetics, molecular biology. We had Josh, but Josh was, at that time, getting to be quite a public figure: writing columns for the *Washington Post* and eventually, two years later, becoming president of Rockefeller University. We needed a more hands-on molecular biologist who was able to work with us on a day-to-day basis. So we recruited two of them, with help from others. I don't know who helped us get them, but Doug Brutlag was one of them. He was in Paul Burk's department in molecular biology. And Larry Kedes in the medical school was a hematologist who was interested in molecular genetics at the time, and was really interested in computational methods. So Doug and Larry were our domain experts, and Peter and Mark were the AI students, and I was kind of the AI and computer science glue for all of that. Then later on when the product of our work turned out to have a very broad applicability because the field was exploding late in the '70s, exploding from a few people to hundreds of people who were interested in doing this. The software guy could be very widespread, first on an NIH-supported timesharing machine at the medical school and connected to what was then the ARPANET. And later, a basis of a company called Intelligenetics that was started by myself and Peter and Doug and Larry.

Knuth: I want to go into companies later. But I guess now would be a good time to just ask about Sumex and ACME.

Feigenbaum: Good point. In fact, Josh and I yesterday were just talking about the history of all that. In 1965 there was a confluence of events in computing occurring, besides the formation of the computer science department which occupied the mind of George Forsyth very much in early 1965. There was a confluence of other computing events going on. Basically, a new generation of computing was arriving on the scene in the form of either the IBM 360 set of machines for some people, or the CDC 6400/6600 set of machines. Even at the time the original PDP-6 was showing up, at that time, it wasn't as powerful as some of those other machines, but it was a very flexible, beautiful architecture machine.

Knuth: This was the first, well, I don't know, silicon revolution.

Feigenbaum: That's right. Solid logic is what IBM called it. So it was time to augment the computing capacity of the Stanford Computer Center. It had a Burroughs machine. It was going to get something else. It could maybe get another Burroughs machine, or it could get something else. George didn't want to handle that. He had his hands full with the computer science department, so he asked me if I wanted to basically run the computer center and handle that. But it turned out that Bill Miller had arrived at Stanford the same day I arrived, January 1st, 1965, as another faculty member in our department, and Bill needed to do the same thing for the SLAC. He was brought in to do that for the SLAC. They needed a massive machine, so it looked like we ought to do whatever we did together because software was a big cost. Software was a big pain in the mind, pain in the neck. We'd better do it together, and we can master one operating system and not two different ones. So we decided to that. Then Josh Lederberg had the idea that he wanted to bring computing to the medical school, and why didn't we (A) go out and get some money to do that from NIH, and (B) join it in with this other plan. That's just what we did. It ended up -- I won't go through the gory details of how we ended up this way -- but it ended up being a 360 mid-sized machine originally slated to be a model 67. But that was never delivered, so it was a model 65 . A model 91 for SLAC, which was a very fast machine. And a model 50 for the medical school machine, partly because it fit our budget, but partly because it was a machine that IBM was selling with -- would you believe this -- I don't know exactly what they called it, but my word would be a megamemory. Namely, you could buy one megabyte of memory. Amazing, one megabyte! You could get one megabyte of main memory, which means you could write a memory-based timesharing system without a lot of swapping. Imagine, you had one megabyte to do it.

Knuth: I can't imagine. I had 10K…

Feigenbaum: Exactly. So we bought a 360/50 with this big memory, so-called big memory. I hired a friend of mine from Berkley, Gio Wiederhold to run that facility and develop that timesharing system. That was called the ACME System. I think ACME stood for -- I'm not positive of this -- but something like A Computer for Medical Education or something like that.

Knuth: And that was different from Sumex--

Feigenbaum: No. ACME was a timesharing system for the medical school, not just for our AI research. In fact, for our AI research, it wasn't all that good because it didn't have well-developed Lisps on it, so we were using McCarthy's ARPA-supported PDP-6 and PDP-10. But then our projects really grew up. Dendral was really successful. We moved it to medical diagnosis and a couple other medical-related projects. It looked like it was time basically for us to do what McCarthy had done up at his lab, but to do it for our own lab. We were on campus, not up in the foothills, largely because we were collaborating with experts on campus. Driving up to that facility [SAIL] was a big pain. We needed them when we needed them. It took a half hour to get around campus and up on Arastradero Road. So Josh and I applied for some money from NIH for a machine like McCarthy's but using the TENEX operating system rather than the one that they were using at SAIL. And we got our ARPA supporter, namely [J.C.R. Licklider].

END OF TAPE 1

Feigenbaum: Let me talk about Licklider a minute.

Knuth: Good, yeah.

Feigenbaum: Licklider had come back to Dartmouth for his second term, second time around. He was the guy who sponsored McCarthy in the first place at Stanford, and sponsored me at Berkeley in the first place. But he had come back to Dartmouth for his second round. And Josh and I -- of course, we had been involved, my group had been involved, with Dartmouth PI meetings and the growth of the ARPAnet at the beginning. We asked Licklider if we could attach a non-DOD machine to the ARPAnet, namely an NIH-sponsored machine. And he said, “Of course.” So we got ourselves an IMP [Interface Message Processor}, I believe it was, at the time, and attached ourselves to the ARPAnet.

Knuth: What year are we talking about?

Feigenbaum: Like, roughly 1973. Something like that.

Knuth: I see, so-- 1973.

Feigenbaum: Licklider came back in ’73, I think.

Knuth: Oh, okay. So then was this the only non-DOD machine at the time, you said?

Feigenbaum: At the time, it was the only non-DOD machine. But that suddenly opened up the prospect that the stuff we were doing -- I’m sorry, to be more specific, the software we were developing in our group -- could be made easily available to a national community of researchers wanting to do AI in medicine. That became the acronym AIM. AI in Medicine was the national group. SUMEX was--

Knuth: SUMEX hyphen AIM, <inaudible>--

Feigenbaum: Yes. SUMEX was the Stanford University Medical Experimental System. Dash AIM, the Artificial Intelligence in Medicine National Group.

Knuth: Do you remember what the internet would, I mean, how many different operating systems were there? You say you have TENEX on this one, and McCarthy had his, I don’t know. It was--

Feigenbaum: Homebrew, yeah.

Knuth: Homebrew, LOTS [Low Overhead Timesharing System]. And then there was ITS at MIT, the Incompatible Timesharing System.

Feigenbaum: Yeah.

Knuth: I suppose there were quite a few different ones then <inaudible>--

Feigenbaum: Yeah, and maybe every machine had a… Though eventually TENEX kind of took over on the DEC machines, but--

Knuth: It's good that the internet actually started with all these different kinds of machines, because that made it more likely that it would be useful in the future.

Feigenbaum: I think they had that in mind from the very beginning.

Knuth: Yeah, <inaudible>.

Feigenbaum: Then we hired Tom Rindfleisch from Cal Tech, a real expert who just ran things. Ran that facility for us. Did a superb job over a 20-year period.

Knuth: Was this used by people all over the country, then?

Feigenbaum: All over the country. In fact, the software I was telling you about before from MOLGEN, when we provided it -- a tool for sequence analysis, which is now common, now everybody does sequence analysis. But it was just us and a little biology group in Wisconsin, I think, that had developed software for it. We had 300 users of that coming in over TIPs, [Terminal Interface Processors] not IMPs, but coming in over TIPs, located all around the country. I remember had put in [?] ARPA had, but in modems on computers called TIPs, so we had a big user community.

Knuth: So IMPs, yeah. It's like I was using Maxima at MIT at that time, <inaudible> system. I think people would, from MIT, would like <inaudible> or people from Wisconsin.

Feigenbaum: Yeah.

Knuth: Yeah, right, okay good, <inaudible> yeah.

Feigenbaum: Russell Amirel's [ph?] group at Rutgers was another example. Carnegie Mellon people.

Knuth: Yeah. Okay, good. Now, let's see. I don't know if I should try to go smoothly through all of these topics or… Well, let's talk about companies now.

Feigenbaum: Wait, Donald. Can I just back up a second and say one more thing about SUMEX?

Knuth: Yeah.

Feigenbaum: SUMEX is in danger of being sort of bypassed by history in the sense that Tom, who is now retired, I don't think he's had an oral history. I don't think the history of SUMEX has been written, but someone ought to go back and do it. There's some absolutely brilliant things that happened at this medical computer facility. For example, the Computer History Museum shortly will have an exhibit in which they're exhibiting the history of the first router that was done at SUMEX. It wasn't done by Cisco, no matter what Cisco, the founders of Cisco claim. That stuff was taken from what Bill Yeager did. And the Computer History Museum is cleaning up that piece of history, but there's a lot of other stuff that someone ought to go back and do that history sometime.

Knuth: Te archives from McCarthy's computer, all the DART tapes, all the backup tapes, have been preserved. Is there a similar thing for SUMEX like for people who'd like to look at the data?

Feigenbaum: Yes and no. Tom Rindfleisch saved 1300 dump tapes when SUMEX went out of existence. And he saved two tape readers, and he's using a PC-based simulator of what the machine used to be. But on the other hand, Tom's pretty busy doing other things, and there's no…

Knuth: He has been the last year.

Feigenbaum: There's no volunteer group that's helping Tom preserve the history of SUMEX. He did give Lederberg and me what amounts to the email of SUMEX from the time we got online to about 1980. 1979.

Knuth: Email, okay, yeah.

Feigenbaum: But then, there's old files in the directories. There's a whole history there.

Knuth: Yeah. I guess I should talk a little bit about history then, since we're in that… I'm trying to segue way through my question.

Feigenbaum: Yeah, sure.

Knuth: You should show everybody the tie you're wearing. It's a core memory tie. I don't know if it's possible to zoom in on that <laughing>, but tell me the story.

Feigenbaum: It's from the Computer History Museum. Actually from The Computer Museum in Boston. Gwen and Gordon Bell gave me this tie once. It's a representation, on the tie, of the wires and the cores that used to go into core memory.

Knuth: In general, you've got this great interest in history and preservation. I wondered if this is late in life, or if you had this all the way through -- if you loved history when you were younger the same as you do now. What are your thoughts about these things? I know there's also talk about semantic web, which we can get into.

Feigenbaum: When I was younger, I was too busy for history and not cognizant of the importance of it. I hadn't lived long enough to understand the importance of it, even though history as a subject was okay. I mean, there was nothing I disliked about it, but it wasn't like I had this lust to be a historian and accidentally ended up a computer scientist. But after I interacted with Josh Lederberg for several years, he taught me the importance and value of saving documents and annotating them and putting metadata on them, making sure they were organized. They were sure to be valuable someday. I began to understand that myself, as I saw ideas unfolding, and influences on those ideas unfolding. As I got older and began to see my own career unfolding, and began to realize the impact of the ideas of others on my ideas, so I became more and more of a historical buff. Then Lederberg and the National Library of Medicine collaborated on the first of what became a series of websites on the National Library of Medicine computer called "Profiles of Science." That convinced me to get very serious about -- well, I was semi serious about -- the archives. I've spent a lot of time on the Feigenbaum archives in 1985 with Henry Lowood, and setting up this collection at Stanford. But more recently I did a second act session and became interested in the digitization of them, [and] the organization of that digitization so that it would be meaningful to historians, curators, scholars, 50 and 100 years from now, who were interested in the flow of ideas around this time. Look how many people are interested in the flow of ideas around Newton's time. Newton enlightenments and the calculus, and…

Knuth: I was reading last night about Darwin. It's not as well-known as it should be, but he was in poor health throughout most of his life. He referred to this all the time in letters that he wrote and so on. There's a wonderful article that just came out by two doctors in Chile diagnosing his medical problems. But it also fits in with all of his life very perfectly. Without this kind of documentation, so much bad conjectures would be flourishing. Now we know a lot more about the way important things were discovered, because-- And of course <inaudible>.

Feigenbaum: Don, another way to look at it would be to say that -- to go back to the very first question you asked me today -- if you're interested in discoveries and the history of ideas, and how to manufacture ideas by computer, you've got to treat this as data. This is data. This is fundamental data. How did people think about what was out of the environment? What alternatives were being considered? Why was the movement from one idea to another idea seen as preposterous at one time and accepted at another time?

Knuth: And you're working now on some kind of a facilitating automatic processing of this data?

Feigenbaum: Yeah. I'll tell you about it a little bit, but I always like to talk about things that have been done and are, you know, someone can… In the future, I want one of these scholars to be able to say, "Yes, he talked about that, and here's the program," or "here's a representation of it." This is just a conjecture. We're working with the Stanford Library. I am working with the Stanford Library on trying to do at least a semi automatization, or at least a very significant facilitation, of the kind of work that the National Library of Medicine did with Lederberg to create this superb historical digitized website. What do I mean specifically by it? I mean, that everyone who is a professor emeritus… I mean, we're going to start with computer science, because that's closest into what I know about. I'm going to hopefully get

McCarthy involved with it, and maybe yourself, other people in computer science. But I'd like to be able to do it in other fields as well. To give every professor emeritus a kind of a workstation, a history workstation, where they have a computer and some software and a scanner. And the help of a library archivist, to get them started with metadata, and get them started with layering on some additional concepts from their field. The buzzword in AI, which I don't like, is really ontology. But lay on some ontology things. Then maybe a graduate student or two will add some inference methods on top of that, and be able to allow a scholar to draw conclusions from what's in the digital archive, as well as just gather data from the digital archive.

Knuth: Yeah. We'd have to find a graduate student who also appreciates history at that point in their life, right? <laughing>

Feigenbaum: This has been a hard problem. We've been looking around. We finally found an undergraduate in computer science who's doing a minor in history, but it's going to be just as hard as finding a graduate student who is interested in chemistry.

Knuth: Well, it turned out that last year was the 300^{th} anniversary of Leonhard Euler's life. Two graduate students at Dartmouth got the history bug, and they went through and they have set up a website now. They digitized 880 Euler papers and his correspondence, and they wrote a description of how… They just loved this work. Once you can get them started on it, there's hope, I think.

Feigenbaum: Can you imagine what that would be like if Euler were alive and you could actually get audio clips from what Euler thought about some of those documents, and how he got these ideas, and who influenced him and--

Knuth: Well, of course, he's one of my heroes, and I'm afraid that I don't want to be disappointed either. <laughing>

Feigenbaum: Yeah, well, that's like Newton. He wasn't a real good guy, but--

Knuth: I know. Apparently, he smoked like a chimney <laughing>. Okay. Great. I think we can only hope that people who are watching this now will see that they should also do the same then for the history of their time, as we're trying to do now, with the new tools that will be available by then. Now I guess I really want [to do] the most cross-cutting of all. I might as well get to it now. What things make you the most happy? This is a hard one, but if you'd sort of look at your whole life and say these were the happiest times. What are the? I don't like top ten questions usually.

Feigenbaum: Yeah.

Knuth: But what does this trigger in your mind when I ask?

Feigenbaum: Great question. Great question. I don't know if I'll get the order exactly right, and if I don't, I'll edit it into the transcript.

Knuth: Edit it later, right. <laughing>

Feigenbaum: The number one thing of all is my wife, Penny, the wonderful person to be with. The period of time after we met and got married was one of the most productive times in my life, because it was one of the most bubbly happy times. It sort of continued that way throughout life. That is the number one thing that makes me happy. The number two thing that makes me happy are those four wonderful girls we have, and now all the grandchildren coming, and they're terrific. The girls all turned out to be wonderful and productive and happy. And so you know what? I hate to tell this story in a world in which lots of bad things happen to people, but none of those happened to us. We get together routinely as a family. Not all the time, but at least once a year we all get together, and then subsets. That made me very happy, has made me very happy and continues to make me very happy. The third thing is relationships with a small number of superb people that I've had, that continue on. Herb Simon is dead now, but right up until… I didn't know Herb was dying at the time. I mean, he didn't tell me that he had a cancer and that he was dying. But I would go routinely to Pittsburgh to see Herb up until the very end. When Al Newell died, I was in close contact with Alan. I happened to be on the East Coast this week, and it was essential, in my mind, that I go and see Josh Lederberg at his place. He's not feeling so well these days. I went, and we talked a lot about not only the past but also is there anything we could still do together. Those things are very… And not just with older people, too. Like, getting together with Doug Lenat is always a great pleasure for me, or Mark Stefik or Peter Friedland or people like that. Randy. Ted Shortliffe. Bruce Buchanan I see fairly often, because I make a point of it. I really want to see Bruce. So that makes me happy. Then, another thing that makes me happy is-- I mean, those are the people things, and the people things are more important than the--

Knuth: That's right. But we can talk about different orthogonal dimensions.

Feigenbaum: Yeah. And so the other thing is--

Knuth: It doesn't have to be a linear order.

Feigenbaum: Yeah. Maybe it's not a linear order. But it makes me very happy to have the feeling that I got to understand very well, and could explain, a very hard problem. Which is: the problem of what is it that makes something or someone intelligent. I don't understand everything about it, nor does anyone else in AI, nor have they ever. But they will eventually. I know enough about it now to feel very good about where we got to in that. That means that it wasn't all dribs and drabs, and it wasn't all a waste. It means that I have some ways of answering the question what to do next, because things have worked out in the past, and therefore it's a reasonable extrapolation that if you follow that track, it will work out in the future. I like that. That makes me very happy.

Knuth: Okay. In other words, you gained some answers to questions that you were trying to look at, and in the process you also gained many friends who shared the journey.

Feigenbaum: Yeah. Wait, Don, I wanted to mention one other thing. Two other things. Number one, I never gave up this idea of wanting to plow into somebody else's field. So the physics you and I were talking about today, that's another example, which is a hobby of wanting to find out what the latest is in quantum cosmology and theories of gravity and so on. And not just that, but it's sort of everywhere. On this trip to New York, I took along Lee Smolin's book, *The Trouble With Physics*, which is a very well-reasoned examination of what's going on there. And I also took along Shakespeare's *Henry VI, Part 1*. That's the kind of… And then the last thing is…

Knuth: Okay, if that makes you happy. <laughing>

Feigenbaum: The last thing is my music. Singing with the Stanford Chorus is just… When I do it, there is nothing in the world that's making me happier than that.

Knuth: Right. That was one of the things on my list. I know that you've done this for a long time. Can you say… I know you might have introduced me to Harold Schmidt ages ago. You've been with the Stanford Chorus maybe from the 60's or--

Feigenbaum: Well so, in 1970 -- in the late 60's -- I tried to look for other things to do besides doing the normal thing I would do: the nerdy thing of taking home work with me at night and finishing it up and preparing for my class and going back to teaching it the next day. I wanted to do other things. One day I saw an ad in the *Stanford Daily*, the student newspaper, for those of you who are, 50 years from now, watching this. "Basses and tenors needed for Beethoven's Ninth Symphony. Stanford Chorus, Harold Schmidt, Conductor." I didn't know anything about singing, but I knew Beethoven's Ninth inside and out. I knew it almost by heart, just having listened to it a lot. So I went over to Harold Schmidt and said, "Here's the ad," and I wanted to do it. But if he auditioned me I'd never make it, because I had no experience in singing. He said, "Okay, just go sit in the back of the room and see how it goes." That was 1970, and I've never left. Still in the back, still there, still doing things with the Stanford Chorus. Beethoven's Ninth, of course, was done many times. I even sought out performing it twice with Japanese choruses in 2005. They do a lot of Beethoven's Ninth in the month of December, which they treat like we treat Handel's Messiah. I don't do three things a year anymore at Stanford, but usually one thing a year at Stanford, and then I find things elsewhere. Like next year, Stanford is going to take the Chorus to Beijing to sing Carmina Burana at a music festival preceding the Olympics. Or, I joined up with the Berkshire Festival Chorus and sang Rossini's Stabat Mater in a Porcini Mass at Canterbury Cathedral, or singing the Verdi Requiem with the Stanford Orchestra and Australian Choruses in the Sydney Opera House, just for kicks. It makes me feel wonderful.

Knuth: That's great. I remember one time you came to my house, and we sang Schubert Lieders. Harold Schmidt was there, and I played the piano. Okay. Now, another thing I <laughing> wanted to bring out is that you have always liked gadgets, as far as I know.

Feigenbaum: Yeah, yeah.

Knuth: You were the first on your block to get stuff. When I was working on a little eight-bit computer in 1980, you said, yeah, yeah, you had been playing around with, I don't, Altairs, or whatever. Things like this. This is fun again to program this way.

Feigenbaum: Yeah.

Knuth: You were the first person I knew to have a CB radio. "Fig Tree" was your moniker, as I recall.

Feigenbaum: Yeah, yeah.

Knuth: So, you have a Bluetooth hearing aid now. <laughing> It has been through your life. Is this a cross-cutting thing, would you say?

Feigenbaum: It's getting worse, Don. Know that I'm retired, and I have a lot more time than I had before. I'm getting geekier and geekier. So I get to… For example, I shouldn't be diving down into the details of the Mac operating system, but I'm using a lot of Macs. My wife is using Macs, and things happen. You just got to know about them, and--

Knuth: So I can call on you the next time I have a problem?

Feigenbaum: If you have a Mac.

Knuth: Yeah, well you know I do.

Feigenbaum: I should be hiring someone to do this, but I love the geekiness of it. And part of it is back to when I was… It feels like I'm nine-years-old and ten-years-old again, and proving Euclidian theorems in Euclidian geometry, where it just felt so good to finally find the answer. You didn't know what you were doing. Then you kind of figured it out, and then you proved the theorem, and it works. Getting back involved with details of computing has that nine and ten-year-old feeling about it. It really feels good to kind of figure it all out.

Knuth: Do you think that we're hiding too much of that from people now? I mean, the kids don't have a chance to do this as much as you and I did when we were young. Maybe not.

Feigenbaum: I'm not sure about [that], because we may be hiding that level of it in computing, but everywhere you turn, the levels are showing up somewhere else. They may have the same problem with… They may have geeky issues with their digital music that they generate, and the MIDI channels, and all of that.

Knuth: Yeah, yeah. No, I'm sort of reacting to… Yesterday, I was reading the abstracts of a lot of new books that have just come out. They all seem to be how to use somebody else's library of routines, and you know, how to learn this language, or this other one, and reuse software, instead of write your own software. It's certainly important to be able to reuse stuff, but it seems to me that if your whole life is just to be reusing, it's not much of a life. I'm reacting a little bit to trying to think of how to put more fun into what computer scientists, or people who are hired as specialists in information theory, are supposed to do. I guess developers is another word for it. It seems that you can't do the geeky stuff as well. But I'm not sure if you have any similar--

Feigenbaum: Well, I just think there may be some common rhetorics [ph?] here. That is, what we're doing is packaging up things at this level, and saying these are tools for you to use for being creative at the next level, and the next level up. Maybe they are being that creative, and we don't know it because we're in our 60s and 70s.

Knuth: Yep. This is true. I can feel myself getting grumpier by the moment. <laughing> Good.

Feigenbaum: Don, I just want to make one comment, though, that is another heuristic from what you just said. I know you feel strongly about this, because I saw it when you were doing TeX and METAFONT: people pay way too little attention to the user interface. They're too geeky in the details, and they're not paying enough attention to the human use of these tools, and how to make it beautiful. Beautiful is wonderful, and when you see something ugly, it's really offensive. And a lot of the software interfaces that people use are just truly ugly. I don't know how you can get this across to computer science people. We try a little bit in our department, but we just can't get it across. Maybe it needs something like David Kelley's new design institute to do it.

Knuth: But you use a Macintosh?

Feigenbaum: Yeah.

Knuth: Which is not ugly. I mean, you presumably enjoy--

Feigenbaum: It's good, not excellent. There are serious issues with it, and I'm sure that they know it. But at least they attend to those issues, and maybe they get 80 percent of them right. But what do you do when people don't even attend to the issues, like if you're Sony? Sony has the ugliest engineering interfaces to their beautiful products.

Knuth: Okay. Yeah, the products are--

Feigenbaum: But--

Knuth: Yeah, I see. It's a wonderful artifact, but then the deep-- but inside of it, it's <inaudible>.

Feigenbaum: Well, for example, Walter Mossberg. For you people 50 years from now, 100 years from now, you won't know Walter Mossberg, but he's the personal technology reviewer for the *Wall Street Journal*. He wrote a column on Sony's answer to the iPod, which was actually a beautiful little hardware device. It was an elegant device. And after he goes through and tells you how elegant it is, he says, "Don't buy this, and the reason is," and he has these three words, "Sony's software stinks." And that's the problem. Engineers don't understand what beauty means in an interface.

Knuth: It's certainly a much under-appreciated talent. You asked me to come back to when I said Josh [Lederberg], how could he have evolved? I better get that off my stack.

Feigenbaum: Okay. So here's the thing I wanted to say. For the audience, this kind of discussion can get a little tedious and rambling, but this part isn't. What I feel personally and have said in public several times is that there are certain major mysteries or, put it another way, magnificent superb open questions of the greatest import. Some of the things we study aren't. I mean, if you're studying the structure of databases, that's not one of the big magnificent questions that I'm talking about. I'm talking about the questions in cosmology about the Big Bang. I'm talking about the initiation and development of life. But equally mysterious is the arrival or the emergence of intelligence at all. Stephen Hawking once asked, "Why does the universe even bother to exist?" You can ask the same question about intelligence: why does intelligence even bother to exist? The world would go on just as well without us being here watching it, processing it, understanding it. So why does it exist? And then, how did it get to be there? From what did it come, and then how does it work? Now that it does exist, we have wonderful examples. Fortunately we don't have to invent it. It's all around us. How does it work? Those are the really magnificent questions, and one of them is the evolution of the intellect. That's when I said, "Please ask me that again." I wanted to get to this superb great question. It's why I try to get students to go into AI, or cognitive science. It's a vector into that question. If they wanted to go into cosmology, I would move them off in that direction, too. That's another set of wonderful questions. That's why I go to hear talks by, say, Andrie Linde in the Physics Department at Stanford. Because I think those are also great questions.

Knuth: Good, okay. Actually one of the first questions I wanted to ask, I just was waiting for the right moment. You've accomplished so much over the years. How do you organize your time every day? How do you, you know, you wake up, and? Do you work on one thing at a time? Do you shut your door sometimes? Are you on the phone all of the time? You work in interrupt mode? How do you go through… is Monday different from Tuesday? What kind of scheduling algorithm do you use in your life?

Feigenbaum: What I'm doing now in retirement is a lot different than what I did during the major portion of my career, where I was a lot more organized about when I did what I did, and the concentration I would put into certain things. Right now, it's more like a combination of demand driven and personal interest driven. Like, dammit, if I'm right in the middle of this Lee Smolin book, *The Trouble With Physics*, and I'm right at a great point, I'm going to continue doing that no matter. Well, if I'm coming down to the Computer History Museum to talk to Don Knuth, I'm going to do that. But otherwise, I'm going to sit there and read that book. Or if I get tired of reading about physics, I'll switch over to the Shakespeare, or whatever. So it's more like personal attention driven.

Knuth: So you have a reservoir of a few things that are currently… But you're always working on something?

Feigenbaum: Yeah. I'm always working on something.

Knuth: You don't have too many things on the calendar now as you used to. But you do, of course, have rehearsals and the--

Feigenbaum: Well, the great thing about being retired is not that you work less hard, because what you do is kind of inner-directed. The world has so many things you want to know before you're out of here that you have a lot to do. But it's more like no one else gets to put something on your calendar. That's what I really like. I get to put the things on my calendar.

Knuth: But let's go back then, 20 or 30 years. How did you work it then? Was somebody else keeping a calendar for you?

Feigenbaum: Yeah, always.

Knuth: And so would you have a leisurely breakfast? You would come into work, and you would say, "Now what am I supposed to do?" And every five minutes somebody else comes in the office? Or did you have special ways? Did you reserve some blocks of time where you could get certain kinds of work done? I'm just curious.

Feigenbaum: The honest truth is, now that you're asking me this, I've never really analyzed it deeply. But I would say that there were blocks of time allocated to what I would call "intelligent delegating." That is, if you wanted to get something done in a laboratory, you wanted to make sure the person understood what needed to be done. If you weren't quite sure, you would work it out with them, and then they would go away knowing the kind of thing they should do. That takes time. And then you've got to get them motivated to do it, anyway. So that was part of the time, setting aside blocks of time for that. Another setting aside blocks of time was for intellectual interaction. Like, I just would be more creative when I met with Bruce and Josh together and some of our people, and we were bouncing ideas off the wall. Josh would come up with 100 different ideas. 99 of them wouldn't be right, and one of them would be brilliant. Just kind of working that out. So that was blocks of time. And then there was just blocks of time to do… Young people don't understand that life consists of a lot of stuff you just have to do that isn't all that much fun, but you just have to do it. And so you just have to sit down and do stuff like that. Like when you're department chairman, or when you're running the computer center, or when you're Air Force Chief Scientist, or something like that. There are just things you have to do.

Knuth: Yeah. I guess I often sort of schedule the things that I most dread to do but I'll be happiest when I finish. Provided I'm ready for it. The hardest thing that I'm ready to do, because then I can get stuff out the door. Because there's always stuff that isn't so much fun. But if you keep postponing it, it gets worse. On the other hand, if you look at it from the wrong point of view, you'll say, well then you'll never end up having any fun. <laughing>

Feigenbaum: Yeah, I know. That is the big problem. And one more thing, I think, about the allocation of time, is that I tend to allocate far more time than I notice my colleagues allocating to other people. That is making sure that they… I mean, I really like other people. I really like the people in our department. I really like the administrative personnel in our department. I just want to keep touch with them, and even if there's nothing really to talk about, just go in a say, "Hello" and be friendly. That I just devote more time to that than other people do. And it's not like I program the time. It's [not] like I say that tomorrow morning at 9:00 I'm going to go in and talk to Pete Turner [ph?]. It's just that I just do it.

Knuth: Yeah. Okay, good. I guess this is a good breaking point.

END OF TAPE 2

Feigenbaum: Don, I wanted to add a postscript to what we were just talking about, getting organized and getting things done. If you look at one level above today's activities -- this month's activities and so on -- here's another heuristic for people watching this thing 50 years from now, or even five years from now. You have to have a more global vision of where you're going and what you're doing so that life doesn't appear to be just sprouting in motion, being bumped around from one little thing to another thing. A good example is there is a lot of chain of activities in my life that stretches from about 1976 until about 1990. If you know what was going on, this pattern of activities is a coherent plan, very much like the plan that you laid out to all of us back -- what, I don't know -- more than 30 years ago, 40 years ago, for these volumes. That was a life plan, and it was coherent. You modified it over time, but it was a coherent set of things that gave order to activities. In my case, 1976 was deciding that if AI was going to be -- by "AI" I mean the artificial intelligence part of computer science-- if that was going to be a robust discipline, it had to have a robust collection of knowledge. I knew that with the set of things I was doing, there was no way I was going to write an encyclopedia of all of AI, but it needed to be done. It ended up being called the *Handbook of Artificial Intelligence*, but I needed to get a lot of help from students and research associates and other people who were willing to contribute to it. I guess, now, these days you'd call it a Wikipedia or something like that. But we had a wonderful editor, and from 1976 through 1983 or 1984, we were able to get enough motivation and enough talent together to make a coherent body of knowledge for the field, which tremendously influenced -- not [just] at Stanford, but across the board -- influenced the development of the field. Because people could now buy a book that had it all in it. So that was one thing. Another thing was late-1970s, early-1980s, capturing the opportunity to transition these ideas from Ivory Towers like us at Stanford into companies. That is, getting on board the bandwagon of venture capital people and all those kind of people who want to do this, so that the ideas can get out there as software. That was a variety of companies to get involved with, including in the mid-1980s getting involved with a big company. I was on the board of Sperry Corporation. Another example of it was same period of time, 1982-1983. AI was known to the rest of the world in terms of a few people who were writing grumbly books about AI, like Joe Weisenbaum's rumble about computer power and human reason. Or other people writing about why AI couldn't be done. Well what about a book about what was being done, and not only what the Americans were doing, but what the Japanese were doing? That was called the Fifth Generation. It sold a huge number of copies. You can't imagine how many people have come up to me -- I'm talking about dozens, maybe hundreds over the years -- saying, "Oh, I read that book and that's what got me into AI," or "That's what got me into computer science. I loved it. I was a EE until I read your book."

Knuth: May I add that to the list of things that make you happy, when somebody like that comes to you?

Feigenbaum: Yes. Then later on, Peter Hart. Peter's an old friend from AI. He's been running the Ricoh Laboratory in Palo Alto for quite a while now, Ricoh Computer Science Lab. He came up one time in 1986, after there were quite a number of successes of expert systems. He said, "Why don't you just write about the successes?" So I did. Penny and I went out and did a bunch of interviews with people who had done this successfully, and got Pamela McCorduck to help us put this together into a book.

Knuth: Is this the "Expert Company" book?

Feigenbaum: Yes, *Rise of the Expert Company* is what the book was called. That was a deliberate strategy that gave some shape to what was going on during that period. To get back to what I was saying at the beginning, make sure you don't get into a random [walk?] situation where everything in the world is demand-driven.

Knuth: Right. Good. Actually, you've done very well of anticipating quite a few of my questions that I had to ask you. So that means that I wasn't too far off as well, and that we're in the same bandwidth. But I wanted to talk a little bit about Japan, since the Fifth Generation and Penny have come up already. I also remember that when I first came to Stanford, I asked you who would be a good dentist to go to and you said, "Oh, I have this Japanese dentist, and it's Okamora [ph?]." So we had the same dentist from 1969. That's probably where you met Penny. So I'm just curious as to sort of how your love affair with Japan and how Japan came into your life over the years.

Feigenbaum: To the first approximation, there was no thought and no contact, no cognitive or physical travel to Japan before about early-1970s. Besides meeting Penny, who represented for me a real big link into the Japanese culture; but aside from that, one of the most startling things that… You were talking about me being a gadgeteer before. I happened to see the Sony Trinitron Television, and I couldn't believe that television could be that good. I mean, it was just a revolutionary way-- It's almost like what people regard high definition now. It's another jump. That was kind of a signal to me that something was going on there. In 1972, Jerry Feldman and I went to an IEEE Computer Society joint meeting in Japan, which I think was held biannually. But I wanted to see more about that, and I wanted to make a trip to Japan. So Jerry Feldman and I went around Japan on a tour.

Knuth: That was like three weeks?

Feigenbaum: 1972. Yes, it was something like a three-week trip. It was broad, but not deep. But that allowed me to--

Knuth: What time of year was it? Probably the summer.

Feigenbaum: I don't remember. It must have been some lovely time of year, because there was no physical discomfort of a hot summer. Then there were some on and off trips to Japan - I don't remember exactly why - during the 1970s. But in 1979, I had an inquiry from a person who's a mutual friend of me and Penny. Penny knew him through IBM Research, when she worked there and I knew him through computer science channels -Professor Yomata [ph?] at the University of Tokyo. It was an inquiry about how about if he nominated me for an exchange ward [ph?] of the Japan Society for the Promotion of Science and how about if we exchanged houses. He would live at my Stanford [house]. He'd work on campus on sabbatical, and I would be a visiting professor at the University of Tokyo, and we'd do one quarter that way. So I did that. It was also part of this--

Knuth: Let's get the timing.

Feigenbaum: 1979.

Knuth: You met Penny--

Feigenbaum: In 1972.

Knuth: In 1972 and you got married in 1975 or something like that.

Feigenbaum: Yes.

Knuth: Okay. So you'd been married for four years and this will be your first extensive time in Japan then.

Feigenbaum: Yes. So we lived a little bit outside of Tokyo. I commuted down to the University of Tokyo. In addition to that, it was part of my job to-- The way this fellowship was set up, it was part of my job to be semi-industrial. That is, as a professor, I was supposed to go off and tell industrial people what my ideas were, which I loved. I mean, I love to talk about expert systems and knowledge-based approaches and so on. So I went around, made a lot of industrial contacts too at that time, plus some government contacts through the laboratory, actually, a very good laboratory in Tsukuba. Well, it wasn't in Tsukuba at the time. It was in Tokyo at the time.

Knuth: Tsukuba is? Oh, Tsukuba, the city.

Feigenbaum: It was Electrotechnical Lab. So I made some contacts with them. In fact, I happened to be visiting the Electrotechnical Lab in Tokyo on a day when a typhoon came through. It was the scariest thing. It was the first time, maybe the only time, I've ever been inside a hurricane. Parts of the building were flying off. But anyway, I made contacts there and it turned out that group, the ETL group, evolved into the group that was responsible for the Japanese Fifth Generation project.

Knuth: So this was in 1979.

Feigenbaum: 1979, but--

Knuth: But it's still sort of not quite born yet.

Feigenbaum: Not quite born yet.

Knuth: But you were there and then later on, you wrote this book about it.

Feigenbaum: So I found out more about it later, as it began to evolve. I got an early draft of the plan they were putting together. The plan was very ambitious from an AI point of view. That is, what they were trying to do for society using AI that they were going to invent over a ten-year period. But it was very much, from a software or a theory point of view, like what McCarthy would love - formal logic, Prolog - and from a computational point of view, it matched very much the spirit of the times in parallel computation. So half of it was a large investigation of parallel computation. Half of it was using that to do highly parallel Prolog. They made a big mistake in that project of not paying enough attention to the application space at the beginning. So they didn't really know what they were aiming at until halfway through. They were flying blind five years through the project. Then they tried to catch up and do it all in five years, and didn't succeed so much. I wrote a book on it, explaining not just their project, but also where the U.S. trajectory had been in thinking about these same issues. That was basically, like I said before, the first time that AI came into public consciousness. The way one of the Xerox PARC people put it in a review that they did of the book, "It was Ed telling AI story by using the Japanese in the plot" -- creating a plot around the Japanese to tell AI story -- which is not unfair. It's not an unfair way of putting it.

Knuth: Well, you need a plot.

Feigenbaum: Yes.

Knuth: Okay, and Pamela worked with you on that book, or was that-- Who was the co-author?

Feigenbaum: McCorduck. Pamela McCorduck. Pamela McCorduck is, at that time and still, Professor Joe Traub's wife. Joe met her in my office at Stanford in Polya Hall. Pamela was my secretary then when I was Computer Center Director, and she had been that-- When I was at Berkeley, she was a junior, undergraduate junior. I don't know if I mentioned this in my past interview, but she was an undergraduate junior and she was the first secretary that Julian - not Jerry Feldman, but Julian Feldman -- and I at Berkeley had on our project. So Pamela and I were on each other's wavelength for a long time. By 1982, she was quite a mature writer, had published several novels.

Knuth: Yes, I have a couple of the novels that she wrote.

Feigenbaum: So what I did was talk it through with her. It's part of this discipline problem you were talking about before: just getting it done by working with other people, collaborating, rather than just sitting down and doing it as a loner. Pamela and I would sit for hours and talk about this.

Knuth: Do you still visit Japan frequently?

Feigenbaum: Yes. I visit Japan now two times a year. In fact, I'm going to be going on May 20th. For those people 50 years out from now, this is 2007. I'm consulting for the Air Force Office of Scientific Research that has an office in Asia, headquartered in Tokyo to sponsor research in Asia. I'm their consultant on computer science and AI research. So I do two weeks two times a year. We used to do a little more often when Penny's mother was still alive. We'd go back there--

Knuth: She doesn't have other family--

Feigenbaum: Her sister and her younger brother live in Tokyo.

Knuth: Okay. Now, let's see. I have a couple of important ones.

Feigenbaum: I want to back up just a second. I want to say two things about the Japanese and my interest in them. One is a tremendous admiration for their skill, intelligence and diligence. It's just really superb. The other thing even bigger is my feeling about -- this is a very fuzzy thing to say, but I don't mind being fuzzy -- the Japanese aesthetic. You can just pick it out anytime you see it. If it's not there, you know it, and if it's there, you love it. It's for me. The Japanese aesthetic is just wonderful. So it's a pleasure to browse through Japanese literature, to look at Japanese designs, to visit things in Japan. It's not that all of Japan is that way. Some of it is just a big bloody mess.

Knuth: You were talking about the new iPod competitor, for example. It had elegance in some parts of it and not in others. It's really hard to-- I mean these are really orthogonal types of expertise, but I know what you mean. That's what drives a lot of our work too, in science. We have the same aspects of taste and stuff. You can't expect everybody to have it all. That's why the world is so interesting. because we get lessons from all the different cultures and we can celebrate that. I wanted to talk a little bit about a phase that we haven't said much about, and that is the work that you did in the 1980s about the "blackboard model" in connection with computer design. Just to put some reminiscences about that aspect and how does that fit in with the Fifth Generation, if at all, or with the other things we've been talking about.

Feigenbaum: I'm going to start back with what AI people call the "blackboard model". Those of you out there are going to have to go look it up, because there's no time to tell the story here in detail, a technical story. But you can find the story in a 1990 book, roughly speaking 1990, by Robert Englemore, called *Blackboard Systems*. It's a collection, *Blackboard Systems*. AI people have a variety of underlying problem-solving frameworks that they use. They combined a lot of knowledge about the domain typically with one of these underlying frameworks. These frameworks can either be forward-chaining kind of frameworks -- sometimes called generate and test -- or they could be backward-chaining frameworks, like "Here's the theorem I want to prove and here's how I have to break it down into pieces in order to prove it." I was doing, at DARPA's request -- again, for those people way out there in the future, DARPA is a U.S. defense department research agency that was funding a large portion of computer science research throughout the 1960s, 1970s and even up to today. They asked me to take a look at a military problem, a defense problem, not a problem of attack but a problem of defense, which had to do with defending against quiet submarines in the ocean; basically, finding those submarines. Where were they? The problem was that the enemy submarines -- in those days, the enemy was the Soviet Union -- Soviet submarines, they're pretty good engineers in the Soviet Union and they were designing their submarines to be very, very quiet and so did we. But the ocean turns out to be a very noisy place. So the signal-to-noise ratio out there in the ocean, listening with hydrophones to the ocean -- it was very hard to interpret in terms of where the submarine is. If there was a submarine, where it is? What's it doing? I didn't think I could solve this problem, but you can't say no to DARPA. So we headed into it. Penny, my wife, was the chief engineer of the project. It was not at Stanford, because it was classified. It was right off campus, in a little contract firm off campus. I tried the same hypothesis formation framework that had worked for DENDRAL, and it didn't even come close to working on this problem. Fortunately, the Carnegie Mellon people, Raj Reddy and Victor Lessesr and Frederick Hayes-Roth, had come up with another framework, which they were using for understanding speech -- it was the DARPA National Speech Understanding Project -- called the Blackboard Framework. It never worked for them. But I picked it up and adapted it for our project, and it worked just absolutely brilliantly. It allowed an arbitrary combination of top-down reasoning and bottom-up reasoning from data to be merged at different levels, which, to use terminology we were earlier talking about, are really levels of abstraction in the problem. There would be experts in the world who were experts on how to go from this level to that level, but didn't know anything about the other level. This is a world composed of a lot of experts. In the program, we call them knowledge sources. Each knowledge source would have kind of a human name for their body of expertise. There were people in the speech world who knew about phonemes, and there were other people who knew about pragmatics, and there were other people who knew about semantics, and there were lexicographers and grammarians. These were all individual experts, but what we did was put together, using this so-called blackboard framework, a collaboration framework where they could all collaborate on a common solution, which was being built on, of all things, a place called the blackboard, which was a commonly available data structure. With all the reasons why each knowledge source gave for its evidence for why something was valid at a certain level of abstraction--

Knuth: It is similar to what people use on video conferences now, where they have the software. People are adding to-- I guess it's a whiteboard now. But it's sort of a display that all participants are viewing and erasing on.

Feigenbaum: That's correct. The difference there would be, in what you're talking it's all people, not programs. We nailed it down a little less fuzzy. We're a little bit more structured and analytic than you would find in these video conferences, but it was a similar idea. We actually produced a program that in its test version, prototype version-- No, I'm sorry. Scratch that. Its prototype version was so promising that DARPA continued it for another two years. They made the prototype version into a solidly running version that was tested by MITRE Corporation against the behavior of humans doing the same thing on the coastal stations of the West Coast. The program did better than people. It doesn't mean anything to the Navy. The Navy just couldn't cope with it and they never installed it, but it did the job. So then we used it, but it was classified. What can you rescue from a military classified situation? Well, the only thing the government cared about was the classification of the frequencies of noises being made on Soviet Submarines. They didn't care about the blackboard framework. That was computer science. So we rescued that in two ways. One was some papers that were written in *AI Magazine*. Penny wrote two really much referenced papers, and we did a kind of hypothetical: how to find pandas in eucalyptus trees. It's very, very difficult to find a panda in a eucalyptus tree. The other thing was we started another big application, which was in the non-classified area.

Knuth: Isn't it supposed to be koala bears?

Feigenbaum: Oh, I'm sorry. No pandas. Koala bears. You're right. Koala bears in eucalyptus trees. Right. How to find a koala in a eucalyptus tree, which arose in a trip I took in Australia and the forest ranger was trying to point out the koala to me and I couldn't see the koala. It was just like a Soviet submarine in the ocean. Then we did a very complicated application, which was one that hadn't been done before. It had been done with massive amounts of computation, but the world didn't have that much computation. You needed a kind of heuristic method for doing it, knowledge-based method. That was x-ray crystallography: interpreting the electron density of crystallized proteins in terms of what atoms were attached to what atoms in what geometric structure in this crystal. You needed not only knowledge about the physics of electron density, which was one blackboard plane, we had also had a chemistry blackboard plane because you needed knowledge of chemistry as well as that. It was the first time that there were what we called intersecting blackboard planes. It was also another very successful application. So now time-wise, we're up in the late-1978s/early-1980s. Now, the _______ [ph?] guys were calling for working on parallel computing. Gordon Bell had started a company with parallel processors. I had talked to Gordon quite a lot about how to structure that. There was another one that Ben Wegbreit was heading out here in Silicon Valley, which combined with a third one, I think, to make a company called Stardent. Anyway, there was quite a movement towards.. Oh, and Thinking Machines [Corporation].

Knuth: Are you talking about general purpose parallel machines?

Feigenbaum: Yes, general purpose parallel machines.

Knuth: But Gordon Bower [ph?]-- Okay, but the psychologists are--

Feigenbaum: No, no, Gordon Bell.

Knuth: Gordon Bell, excuse me. Okay.

Feigenbaum: Not Gordon Bower. Gordon Bell had left DEC and started--

Knuth: Yes, and he went to NSF about this time?

Feigenbaum: He did at some point, but then we started a company.

Knuth: In promoting parallel--

Feigenbaum: In parallel machines.

Knuth: Right. Okay.

Feigenbaum: When I looked at that, I said to do AI problems, the blackboard framework was ideal. Because the activities of the different segments of computation could be, not totally but largely, parallelized by these different knowledge sources working on different pieces of the problem -- either communicating to a common blackboard, or distributing the blackboard to all processors all the time, an updated blackboard. So we started down that track. DARPA liked that idea. We borrowed a computer designer from DEC, Bruce Dilaggi [ph?], and we had Harold Brown, one of our really good people, and mathematician working on the problem with some graduate students. We were simulating parallel machines, the reason being that I had no interest in building anything. In fact, I didn't even believe in building anything. I believe you ought to test out the ideas before you build it. Building is expensive, and if you got the wrong thing at the end, you'd be very unhappy. Well, it turns out that in that particular game, namely the game of architecture of machines, if you're a digital systems lab -- what do we call them? -- CSL persons, computer systems lab person in our department or similar departments, you don't have any credibility at all if you do it at the software level. If it's not at the hardware level, it doesn't exist. So it was almost impossible to communicate ideas to these people. First of all, they weren't interested in AI things anyway. Second of all, it turned out that, with Moore's Law operating, there was plenty of power anyway for the AI methods to work on ordinary machines. You didn't really need parallel machines. So it was an extremely interesting endeavor, but it didn't really come to anything then. But now, we're running out of gas on Moore's Law. Now you're finding Intel and AMD are putting -- it used to be two cores on a chip. Now it's four. It's going to be eight. They're talking about 16. That was exactly the range of multiprocessor activity that we were doing then. We weren't doing Thinking Machine stuff with a vast array of 10,000 processors each doing little things. We were using a few processors doing big things: knowledge sources for a blackboard. I'm getting the feeling that we have to resurrect some of those papers -- like doing a volume of selected papers from that era -- to resurrect those ideas. Long story.

Knuth: Right. No, that's good. It's another case where something happens before its time. But it doesn't mean that it has to die forever. Very few of the things that have ever been discovered by computers turn out to be never useful. They're always dominated by something else. Small changes in hardware capability can mean that all of a sudden, something is much better than the other thing, which used to seem to be better. Now let's see. I guess I should ask something about the experiences of starting up companies, because that's a big thing, although that's not one of my personal interests. I'm doing it only because I know there are people watching who are interested, more interested. It's important for people to make a living and all of that. So I have to admit my own bias is less in that ___________[ph?] but still, that's my own problem. I know that there's Intelligenetics, IntelliCorp, maybe more companies. You mentioned _____________ [ph?] where you serve on the board, but I'm more interested in the startups I think. Some of the stories associated with that, and how you got the idea to do it. Did it come during the night or whatever?

Feigenbaum: So that also is a packeted [ph?] story. So let me start by saying that there were really two inspirations for starting companies. One was kind of a general inspiration, and the other was quite specific. The general inspiration was wanting to get your ideas out there, and along the way, making some money on these companies, just like some of us do consulting. It was that sort of thing. Not only for yourself, but also involving other people--

Knuth: Nobody is rising to the occasion and the world needs something. I had the similar thing with fonts and so on. I didn't mean to--

Feigenbaum: No, no. So let me head down the story. Then the specific inspiration was that there's a Stanford culture that-- It's an order of magnitude stronger today than it was at that time, but at that time, it was still very strong, in 1979. It sort of came down to the Hewlett Packard life story of people starting companies It was the Varian brothers being professors at Stanford. Started Varian, and Hewlett and Packard. Earlier than that there were several stories, even smaller stories; Elliott Levinthal starting Levinthal Medical Electronics, and maybe a dozen other companies. But right about 1979, there were a group of companies that were being born in connection with the emerging biotechnology, out of the medical school. In fact, so many companies were being born that Don Kennedy, who was president at the time, started to worry about whether he had to have special regulations at Stanford to regulate conflict of interest between faculty and the companies. That's how many companies there were.

Knuth: When did Djerassi--?

Feigenbaum: Oh, Djerassi's work was never part-- Djerassi came here to Stanford as a joint -- that is, a 50% appointment in chemistry, 50% doing his company, which was Syntex. It was early on that Stanford and Djerassi got together on the great idea to put Syntex in the Industrial Park rather than in Mexico City or wherever it had been born. Do you want to switch tapes now?

END OF TAPE 3

Feigenbaum: So in that atmosphere, in that environment, we saw two things unfolding. One was that there was a very large demand for a piece of software we had produced in the heuristic programming project, which was the software generalization of the MYCIN medical diagnosis expert system called EMYCIN; very large demand for that. And a smaller but substantial demand for a blackboard framework piece of software that we had developed called AGE, which modestly stood for Attempt to Generalize. We were attempting to generalize what we had done in the sonar and in the crystallography examples. That was one instance where there was a lot of what you might call "user demand" for this stuff. Companies were after us. We were shipping out literally hundreds of copies of the software and documentation. Seemed like a company could do that better than a laboratory. [The] second place where we had a lot of demand was users coming in over the ARPANET, maybe by that time they were coming in on the NSFNET portion of the network, but they were coming in to use the SUMEX machine to do sequence manipulation in the unfolding, emerging sequence databases that were being built at the time. It looked like that was also the basis for another company, because a company could do that. I mean why should we waste our PDP-10 resources serving the nation for this, when they could be buying it from -- a company could run it and they could buy from that company.

Knuth: Was it university researchers, or were these other companies buying the time?

Feigenbaum: A combination of both.

Knuth: When you say “at the time”, I’d like to put dates on them. Which was first, second, third, and approximately when, or did they all start out at the same time?

Feigenbaum: I’ll have to edit this when the transcript comes out and get the exact dates. In 1980 we decided to move forward. But we were in conversation about moving forward on both fronts, the biology front and the expert system software front, in two separate companies in 1980. It was easier to get the biology one moving first, because the only people involved were myself, the two biologists and Peter Friedland. We got another person involved, a graduate student in the Graduate School of Business who was taking a course on how to write business plans. We got the faculty member to target him at us, and he wrote our business plan. We sold it to a venture capitalist who I happened to know because they were friends of Ed Fredkin -- we were talking about Fredkin before. Fredkin had gotten money from them to start -- you remember Perq, the computer company [Three Rivers Computer Corporation]?

Knuth: Yeah I invested.

Feigenbaum: Yeah, I invested in Perq, and so did Raj [Reddy], and so did Herb Simon, and you now. Anyway, Fredkin knew these people, and therefore I got to know them. We got Intelligenetics up and running in, roughly speaking, late 1980, selling sequence analysis software on a DEC machine that was networked in some way. I’m not sure how, maybe it was via the Tymshare net or something else at that time. The other company was called Teknowledge. As you could see the difference in the names, Intelligenetics points to biology and Teknowledge points to technology and knowledge based systems. There [it] was not 4 people or 2 people, it was 20 people. Because my view of that was that everyone who had helped us get to this point needed to be, in some sense, rewarded by being a participant in this new enterprise. So I just made a list of all the people who helped us, who contributed something. Like the graduate student who did EMYCIN, and Tom Rindfleisch who helped us with SUMEX, and Bruce Buchanan who was at that point, like, associate director of the laboratory, and Bob Englemore for blackboard systems, and Penny my wife, and people like that who had contributed. Plus I got in two other people. One, Randy Davis from MIT, who had contributed a little bit earlier but had left. The other was a very good articulator of the ideas from RAND Corporation, a guy by the name of Frederick Hayes-Roth.

Knuth: And Denny Brown was involved with Teknowledge too?

Feigenbaum: I don’t think so. I don’t think he was. I’m not sure of that, but I don’t think he was, because I don’t think he was a member of the heuristic programming project which later turned into the Knowledge System Lab. Now starting a company with 20 people is bizarre. I didn’t know it at the time. It’s only later when people in the entrepreneurial world who know about these things basically said “Are you crazy, you can’t do that. You have to have a group of people who think in common, act fast. That’s the way the culture works.” I never thought that 20 people would be a hard thing to do. Turned out it’s like herding cats, and it was a lot more difficult than we thought.

Knuth: It’s also a big responsibility. I mean, you’re affecting their lives so directly. It’s not just like being a teacher.

Feigenbaum: Well, I had thought that we would get a professional businessman to run the thing, and the professional businessman would get venture capital and then using the venture capital they would hire good people. Instead, what happened was a significant chunk of our own people gravitated over there, because the work was exciting. It was a new thing, and it was paying a lot more than we were paying. So in a way we kind of decimated our laboratory by doing this. Anyway, long and short story is they spent a lot of money -- they being the professional business bank that we hired tried to negotiate an intellectual property agreement with Stanford for the rights to use the EMYCIN software that had been developed at Stanford. Stanford at the time were very much of a novice in the area of dealing with small companies. They still are difficult, I hear, but they were impossible at that time. The guy just said "Okay, Teknowledge will develop it's own. Forget EMYCIN". They spent a lot of money doing that, and by the time they finished doing that, Osborne had a thousand dollar version of that software -- or maybe a hundred dollar version of that software out -- and running on the early PCs, and their $5,000 version wasn't going to sell. So it didn't work out so well for that company. Eventually it got sold, and then it resurrected itself. It's still in business. It's still in Palo Alto. It still employs a couple of dozen people, doing military applications mostly.

Intelligenetics couldn't get a grip on the biology software market because those people at the University of Wisconsin I was telling you about before, basically were offering the same thing for free. PC's were coming into play, so you didn't need one end of a PDP-10, you could buy your own PC for $2,000 and run the University of Wisconsin software for nothing. So it changed it's name from Intelligenetics to Intellicorp, and started to sell a version of some software. It had license from Stanford that we had done in our laboratory, which was one of the first languages that incorporated object oriented programming with inference methods. It was called KEE, Knowledge Engineering Environment, developed by Peter Friedland and Mark Stefik and a few other people in our lab.

Knuth: Did this run on PC's or workstations? In those days there was quite a difference between a personal computer and a work station.

Feigenbaum: Intelligenetics was close -- this was before it became Intellicorp -- as Intelligenetics, it was close to being Sun's first customer. I don't know if it was Sun's first customer, but because I knew of the emergence of the Sun workstation at Stanford, and I knew what power you could have for a few thousand bucks in that work station, I immediately oriented the company that the business guys go make a deal. That was when Vinod Khosla was President of Sun. He was the very earliest presidents, and Sun was still stumbling around. In fact they weren't really such a good vendor, because their machines weren't that reliable at that time, but we were very early customer. Then Intellicorp, I think they pursued the same strategies to offer it on workstations. But by that time the LISP machines were out, and this stuff ran really well on LISP machines. So they were selling them on LISP machines. They were selling them on the Sperry version of LISP machine, the TI version of LISP machine. They were selling them on the portable versions of a LISP machine, the desktop versions of the LISP machine, whatever they were called -- Micro something or other. They had lots and lots of applications, very sophisticated applications that they were doing. They, too, never got business traction and lasted right through the dotcom bust, until during the dotcom bust they went bust. So they lasted, roughly speaking, 20 years. Intelligenetics got spun off and were sold to Standard Oil Research that wanted to get into molecular biology applications. Intellicorp lasted about 20 years. Teknowledge is still in existence. They're all in the space called the "Living Dead", where they're not big like Google or Yahoo or eBay, and they don't go out of business right away. They just provide enough business to keep their employees employed. There are lots of these companies in Silicon Valley. Then there was a third company which I started jointly with a professor in the Civil Engineering Department at Stanford doing…

Knuth: Who was that?

Feigenbaum: Ray Levin.

Knuth: Because, I was thinking, the Civil Engineering have some kind of a blackboard model that they use for scheduling a job. The electricians put up their data, and the other contractors too.

Feigenbaum: That could be the case, because one of our PhD students ended up running their big laboratory -- John Kunz -- so that could be. I just don't know that. But anyway, one of the customers of Intellicorp in Europe in Finland was using KEE to do engineering design of big, I mean gigantic, boiler systems for huge factories. It was one of their earliest applications, or maybe the earliest application, of AI to engineering design. It looked like it should be just a really great AI application area, where you could sell it for a lot of money into companies that would save a lot of money using this software. It was in LISP and it was programmed in KEE, which ran in LISP. Ran on workstations.

Knuth: When you say KEE, I'm not familiar with that, can you spell that?

Feigenbaum: KEE is Knowledge Engineering Environment, KEE. It was an object-oriented system on top of LISP that also incorporated inference software in its methods. KEE, was actually designed, believe it or not, based on some software at Stanford but redesigned by Richard Fikes who was Vice President for Research of Intellicorp. One of Richard's great contributions was KEE. That company Design Power Incorporated was two Finnish guys who had developed the software, came to the United States, and with permission from their company -- I don't know exactly, I can't remember the grounds for using the intellectual property from their company -- they plus one person from Silicon Valley kind of launched this company. I was one of the founders. Ray Levitt was one of the founders. It had a very difficult time selling into the… Sales that we thought would be so obvious just weren't. There was tremendous amount of inertia in the big engineering companies to adopt this. Eventually the company kind of went partially bankrupt after it's first round of venture capital. It couldn't get another round of venture capital because there was some difficulty with Finnish investors that were in this company. They didn't quite understand the ground rules of Silicon Valley and they basically screwed up. So the original company had to reconstitute itself. But at the end of the day they survived long enough to be bought by another software company about a month ago in 2007. In the end their ideas still survive and the software still survives but not in a big way. I just want to mention that during the dotcom boom, I was approached by several people who wanted to use expert systems in whatever their categories were, in that they were trying to exploit web-based something or other. Would I be a helper, a guru, an early investor etc. I did some of those, and some of them turned out to be extremely good. Like one of them merged with a company which merged with another company, which is now shopping.com. It was expert systems for the shopper.

Knuth: Well that's good segue, unless there's something else you want to say about company. I wanted to focus on expert systems as it interacts with other streams of AI. I know that Nils is going to be interviewing you about this but I wanted to bring it out. Artificial intelligence is a huge subject, and it doesn't only have neats and scruffies. I'm thinking, for example, of the strains like Stuart Russell at Berkeley, and Daphne [Koller]. There was a couple of journals in the library, there's one journal called Heuristics and another one called Heuristic Methods, or something, Heuristic Science. I don't know. There were two journals that start out with "heuristics" in the title. So I was browsing through them the other day as I was in the stacks. But I looked in the issues from 2 years ago or something and it would say… I'm sorry, there's a journal called Expert Systems and Expert Systems with Applications, or something like that, not Heuristics. But I looked inside and, you know, a certain percentage of the papers were about expert systems, but others were about fuzzy logic or genetic algorithms. So maybe this is all of a piece. But of these many currents going on, there are neuro nets, articles in there. I know the Japanese use fuzzy logic for everything. I don't know why, because I've never understood it. So we've got, you know, Stewart Russell's approach is -- well what would you call it -- automatic learning, machine learning. And Daphne's Bayesian nets, there's fuzzy systems, you know, there's Doug Lenat's gathering of knowledge bases, and all these things. So in 25 words or less, how do you view this map?

Feigenbaum: Very, very, very good question. I'm going to do it, but it won't be 25 words or less. It might be 250 words or less.

Knuth: I'd like a guide to the perplexities.

Feigenbaum: Once upon a time there was, to use words I just got right out of Lisa Mullins book on physics, there used to be exact knowledge in our expert systems. That is, when we first started, we started with carbon atoms, oxygen atoms, valances, chemical grafts connecting these things, rules for breaking bonds. Either the bond broke or it didn't break. These were if-then rules for representing this knowledge. We had a lot of that knowledge, and none of it was fuzzy. None of it was probabilistic, or in any way probabilistic. So Ted Shortliffe started working on MICYN, and then it turned out that doctors wanted to represent their knowledge with a little bit of fuzziness. In fact, they were using several English language words that covered the space like "likely", or "unlikely", or a couple of other variants of that in the middle. I forget the details. There were common words in medicine, but there were only like 5 of those. Ted expanded that to be 10 categories, not 5. He could have probably gotten away with 5, but he did 10, and set up a little ad hoc calculus to combine those numbers into what he called a confidence factor. Do a little bit of reasoning in the background with this confidence factor. Now that kind of ad hoc… Oh, we used it, incidentally, on the sonar thing. Perfect. It was just wonderful. It just worked perfectly.

Knuth: Sonar?

Feigenbaum: Remember that sonar example of finding submarines in the ocean?

Knuth: Oh yeah.

Feigenbaum: We used that. Penny and I just used a short list scheme right out of the box and it was great. Let me tell you the next step, and then go back a step. The next step was: that got a lot of people unhappy. If you're dealing with something where the words were "likely" or "unlikely" or [like] that, my god, we mathematicians have invented a whole calculus for that! Why are you being ad hoc about it? Why all this adhockery when you could use mathematics? Frankly that was almost exactly the kind of words that you would hear. To which the answer was, "Hey, folks, you don't bring a big cannon to bear on something where you don't need it -- where a little pop gun would do." Let me go back now, give you an example from Djerassi. Here's one of the world's great minds. One of the world's great mass spectrometrists. We're showing him--- we're extracting knowledge from Carl about some mass spectrum. Mass spectrum has a molecular fragment weights along the X axis, and it has relative abundance of fragments at that weight on the Y axis. We were setting up rules for the fragmentation of certain molecules that were showing itself that way, representing itself in that way. We said "Well, Carl, what are you going to do with these Y axis points?" He said, "Well first of all, let's ignore them." In other words it was zero-one. Either the fragment was there, or it's not. And we said "Wait a second, these are lots of peaks. You're not going to ignore the heights of the peaks, are you really?" He said, "Okay I'll tell you what. We'll just call them high, medium and low." So all that the chemist needed was: either it's there or it's not there, or else the most they wanted to use was high, medium and low -- three values. So went to go back to the thing. We were saying "Well you don't really need… You can set up a very simple logic, an ad hoc logic, to deal with this." Along comes La Vuisada [ph?] and he says, "I'll put in some systematic logic into that". It wasn't actually the Bayesian logic that's being used now, it was more, like a La Visata, it was less ad hoc than ours but it was still pretty ad hoc. The Japanese got to love that. They got to love the idea that there was a little calculus involved, a little mathematical calculation involved in doing that. So the first application that was done in Japan was -- I helped to advise him -- it was done in Hitachi's system development lab. It was an expert system to control the breaking of subway trains in Sendai. You didn't want the subway train to come to a sharp stop, which would jerk all the passengers. There was expertise in the conductor. The conductor would use his expertise to know how to apply the breaks. Well, he needed a little bit of fuzziness there to accommodate certain situations. It got used all over the place, from subways all the way down to fuzzy rice cookers. But these were little expert systems, with 20 rules, 50 rules, but with a little almost ad hoc calculus. Well then came a couple of students of ours, David Heckerman, Eric Horvitz. Remember them? David won the ACM thesis award the year that his thesis was done. Both of them work at Microsoft Research now. They were working with Ron Howard in the Industrial Engineering department on a more systematic decision -- what should I call it -- decision-making algorithm structure. That turned into what are called belief nets and these Bayesian analysis. Then it became really sort of a hot topic. Then you didn't have 5 people working on it; then all of a sudden you had 500 people, or 50 people at least, in the field working on those kind of issues. That became its own little field. Any field like that becomes self-propagating. It doesn't matter if Ed Feigenbaum keeps standing up and saying "You guys don't need that armada." There is no reason for coming out with a number like .69108. It doesn't make any sense at all. These problem solving activities are more or less qualitative in the way Polya would have thought about it. That has no effect on people. It's what I called "physics envy" before. They absolutely have to have it. I look back at the paper Wigner wrote, which was the paper in 1961 or 62 on the unreasonable application of mathematics to the physical sciences.

Knuth: "Unreasonable effectiveness".

Feigenbaum: Effectiveness. Well, even for physics it was unreasonable. Why do you need it for this stuff? And yet that doesn't have any impact. So when you read Stewart Russell's book with [Peter] Norvig, for example, on what he calls "Artificial Intelligence, a Modern Introduction". What he means by modern is, "I'm going to get rid of all that scruffy stuff and I'm going to put all the neat stuff in." Then you get Nils [Nilsson] saying, like he did at the AI Lab retreat two weeks ago, he was saying "I used to be one of the neats, and I wrote a book with [Michael] Genesereth on it. Now I'm moving in the other direction. I'm a scruffy now."

Knuth: Well, I don't think you have to be an either–or, do you? I mean, can't you appreciate some of the talents...

Feigenbaum: Yeah, I think so. And I think that's the way it's going to end up.

Knuth: Of course. I work with font design, and it's a similar thing, where if you look at the people who are getting drawings from a great designer, they will measure everything with a micrometer to see exactly where his ink stroke went. But if you have the designer in your office with you, when Hernsob [ph?] came to visit me, you know, "Oh, this looks like about 1 to 4", and then you just do it. So these decimal points was not where the action was.

Feigenbaum: The question is, will that burn up a lot of graduate student time, and faculty member's time, and pages in the journals?

Knuth: No, but I think the fuzzy logic people would say, "Well, but we really aren't concerned about the numbers so much as the axioms that these things satisfy." Then they can then get a higher-level view of it, once they know that there are some algebraic operations that they can do. Then they can use that as a guide to taking the next step. So I'm saying we don't really necessarily have to… I mean, some people might, you know, wear their neat hat one day and a scruffy hat [the next.] Sometimes they might be both in the same body.

Feigenbaum: Let me tell you one more thing: that Doug Lenat's work doesn't fit into this paradigm at all. Doug's work is orthogonal. Doug's work is whether or not you use any numbers at all to characterize these relationships. Doug's work is to prove that huge amounts of knowledge will explain, or will replicate, common sense reasoning. That there is nothing to common sense reasoning but a huge amount of knowledge. But huge for him -- you know, right now it's 5 million things, and maybe it needs to be 500 million things.

Knuth: How does common sense affect versus experts. I mean, experts is uncommon sense.

Feigenbaum: And experts… So Herb Simon did a series after he said he was influenced by the expert system work to go look at what it takes; what makes a human expert at something. His paper with Bill Chase for, at least in the case of chess playing masters, indicated that the chess masters knew about 70,000 things, plus or minus 20,000 things, of specific chess lore. More than the amateurs knew. And that it takes about 10 years to acquire these 70,000 things. So that's not the 7 million, or 70 million, or 700 million things that Doug is after. It's a narrow thing. I don't know if I told you that story about the pulmonary function diagnostic guy. We worked with a guy. When EMYCIN was born, Penny and I decided we are going to test out whether it's indeed the case that you could shove a new domain into EMYCIN quickly. I had a relationship going with Pacific Medical Center up in San Francisco, and I knew a guy up there who was a pulmonary physiologist who did lung disease diagnosis from a machine called a Spirometer. It's a machine that you breathe in and out of. I don't know if you've ever used it. You go <breathing in and out> and then the curves of your breathing show up, your intake and your outflow of air. From that they diagnose the condition of your breathing apparatus. We worked with him for 3 weeks from the first hundred cases that he showed us from his files. Got a good set of rules. Polished them with test cases from another hundred cases that he had. By the end of 200 cases, we were at the point of diminishing returns. We had it. We nailed it, and we nailed it with 400 rules. This guy looked, and he said "Is that all it is? Is that what I know, 400 things?" And we said, "Yeah, Bob, it looks like it." And he gave up doing that for 5 years. He went and did something else. Because he couldn't stand the idea of it: what he was doing was captured by 400 rules. Well, for some it might be 70,000, but common sense is way, way more than that. Common sense means almost everything you know.

Knuth: What about the… I'm thinking about Rod Brooks, you know, the physical-based intelligence or whatever, the way things, they just are little motors, that are interacting by evolution in ways that aren't understandable by logic.

Feigenbaum: So it turns out Rod himself doesn't believe that anymore. But it's not because he changed his mind, so much it's just that it was obvious that what he was proposing wasn't the whole story, and he's just come to encompass more of the whole story. It's the view that the world is its own best representation. So don't try to represent the world.

Knuth: Embodied, or something, is one of the buzzwords that I heard around MIT.

Feigenbaum: I don't know that. I'm not sure about that buzzword. But the idea is, don't try and capture the details of the world like McCarthy would do in formulae. But instead, just respond to the details of the world in whatever is a useful way to respond to it. So if you're an insect and it's a pebble, responding to it might be to lift up your leg to touch the pebble and then move on. Well all of us… My interest in AI is conceptual reasoning. In fact my goal, I kind of called it when I wrote the article for JACM a few years ago, 2002 I think, trying to redefine Turning's test. I called this goal "Einstein in a box." I really meant it. I really meant that I wanted to replicate the mental qualities of an Einstein, in terms of conceptual understanding, reasoning, deep reasoning, insightful reasoning. I said to Rod, what you're doing is great for insects, you know. You might make the world's best insect, but so what? So it's the world's best insect. Now that's not really fair, because years before, eminent psychologists thought exactly the way Rod was thinking. Like Skinner's stimulus response is exactly the way Rod would describe what his leg on his robot was doing. It felt this pebble, and it responded by lifting up the leg. Skinner, you know, was famous. He won all kinds of awards for believing that you could frame human thinking that way. His view was… It's like a big fight between Skinner and others. I can't remember the name of the psychologist, but he was responsible for a brand of psychology called mental maps. There was a big fight in the 1930's between Skinner and the mental maps people, and the mental maps people won. That is, we do have a representation of the world in our head. It is a symbolic representation. Mice have it, not just humans. So but you can't blame Rod, I mean, that's a viewpoint which has a distinguished background.

Knuth: Well, it still might contribute to common sense.

Feigenbaum: You mean stimulus-response?

Knuth: Yeah.

Feigenbaum: Yeah, sure. You see something and you just fire off on it.

<Crew talk>

Knuth: Let's look now at vectors for the future, and if you have some last words of advice to people. I'd hate it if somebody asked me to predict what the future is going to be. So I'm giving you a choice of two questions here: either to say what you think is going to happen after today, and/or you can also say, besides everything else I've been mentioning, what do I want to tell you people of the year 2050.

Feigenbaum: The people watching this, I'm not going to be able to say this in two minutes.

<Crew talk>

END OF TAPE 4

Feigenbaum: Let me say the answer to your question at two levels: the level of the person who might be watching this, a young person maybe, undergraduate or even a graduate student who has a full career ahead of her or him, and then the person who's watching this thing fifty or a hundred years from now. To the person watching it, to the younger person who's looking for career tracks, I would advise actually, which is another way of saying I would predict, that it [AI] is a good thing to do. First of all, to work on the modeling of--- we're making good progress again after a while, after an interval -- on the modeling of human cognitive processes, that is a really serious goal. It's a goal of AI that it comes and goes. AI was largely engineering for a long time. It started out as largely psychological modeling. There's now a lot of data coming in from neuropsychological modeling, neuropsychological data–gathering, that right now is not of immediate benefit to the modeling but it's going to be. It's a good time to be getting into that field, to try to figure out more of how a brain functions. Now I'm not talking about how the physical brain functions, as much as I am talking about how the mind operates in the machine called the brain. So more work on that area. This is a good time to do it. Another good thing that I predict will be coming along, is in the artificial intelligence, the engineering part of our field, not the modeling of cognition part of our field. The use of AI programs to find complex relationships in the immense databases that are being collected right now. Another way to say this is, it goes back to that original goal of Lederberg's and mine: to do science by machine. I talked to a researcher at UC Davis last weekend at the National Academy who was working on a gigantic data collection experiment in astronomy -- astrophysics -- with his sensors were just pulling in like 50 terabytes a minute, or something like that. Just huge. He was saying "What I really need along side of all this is…", and he was asking me if I could supply it. He says "I need a good guessing machine, because I can't look at all of that data. I need something else to look at that data, and basically come up with what Paul would have called the art of good guessing." We would call it clever hypothesis formation. There's going to be a lot of that. We're not going to get the Einstein in a box, but we'll get a lot closer to the Einstein in a box in the next decade or two. That's a very good problem to work on. Now notice that I'm talking about structural, conceptual relationships among concepts in the data. I'm not talking about superficial relationships such as you see emerging from these Bayesian models, or from the neural network models, Don, that you mentioned before. They don't give you any insight. They just give you a bunch of numbers attached to notes, and that's not an insight. That's not something that another theory can build on.

Knuth: I just wanted to say that even though I'm not in the field of artificial intelligence, that I also can recommend people to go into this. Because all through the years [of] people working on artificial intelligence, from my point of view as an outsider it has great value, because you're working on a very hard problem. Because you're working on hard problems, you're going to build tools that people like me are also going to be able to use. Because in order to solve those problems, you're going to think of something that I wouldn't think of working on an easier problems.

Feigenbaum: Which is true. It happened almost on day one, with the invention of list-processing languages. No one set out to just invent list-processing languages, they set out to do a certain kind of computation and it was just horrendously difficult to do.

Knuth: Yeah, without IPL-V I wouldn't know how to solve all kinds of ________[?] problems.

Feigenbaum: The other thing was McCarthy with his insight about timesharing systems. McCarthy was interested in seeing machines interact with people, because that was in his artificial intelligence genes. That's what he wanted to do, and there ought to be a way to do it, and he thought of timesharing as a technique for doing that. I think for the people who are way out there, the fifty to a hundred years or so people, this is a game. What I'm going to propose is a game that I'm not going to win. You are going to win because you are going to be alive and I'm not going to be alive. But it's the question of: will we have fully explicated the theory of thinking in the time of your lifetime, not my lifetime? Will all the questions be answered? Will we understand it? We will understand all the processes that go on, in the same way that physics strives to understand. When they talk about the standard model in quantum mechanics, they really mean that they've got it nailed; they've got virtually all the questions answered. Now of course there are mysteries and mysteries and mysteries below that. But it'd be very interesting to see what you people of a hundred years from now know about all of this. A hundred years is an enormous amount of time in scientific time. What I do feel about computation is that we have gotten into the habit of depending on exponential growth, and the amount of computation bang for the buck that we get of our computers. Once we're in this habit, we're not going to lose a habit. We may say that we're running out of gas on Moore's Law, but someone's going to find some other paths. Like for example, recently we may have slowed down on the Moore's Law side, but we're doing faster than Moore's Law on the storage side. We've made some trades in computation in favor of storage, and we're going to continue to do that. So we're still going to have the amazing wonderland that we've had over the past few decades in computing. Now the question is conceptualizing: are we going to be able to get the big picture right? You guys will know, and we won't.

Knuth: All right. Well, too bad we can't be with you. But anyway, we wish you all the best.

<Laughter>

END OF INTERVIEW

