This is an interview with Eric Roberts of Stanford University. It is being conducted by Barbara Boucher Owens [on July 6th, 2009] in Paris, France, as part of the Computing Educators Oral History Project. Did we get your name correctly?

E: You did. Thanks.

B: We start way back at the very, very beginning. Okay?

E: Okay.

B: All right. So tell me a little bit about your family.

E: Well, I grew up in an academic family. My father was Professor of Political Science. He travelled to a number of places, but he ended up getting his permanent position at the University of Nevada in Reno. And so I grew up in Reno, and later the family moved to Carson City, which is the state capital.
There are a number of other people in the family that I got to know over time who were either teachers or filmmakers. They were in Hollywood, so that was sort of where I drew my inspiration from. They were all quite intellectually engaged and so reading was a part of the family history. My mother’s sister, for example, during the time that I knew her best, read three books a day. And I try to read three books a week and everyone thinks that this is wildly crazy to try to undertake that much. And sometimes … one can do even better.

We were also very active politically as a family. My grandmother was a Quaker anti-war activist in the period and worked with the American Friends Service Committee doing reconstruction work in Germany after the Second World War.

**B:** Is this maternal or …?

**E:** My father’s mother. She went over in 1947, and the American Friends Service Committee won the Nobel Peace Prize for that work and so … she was only a small part of it, of course, but that was the kind of thing that I grew up with. And she was extremely a strong influence on me, as certainly my father was as well.

**B:** Did you have siblings?

**E:** I have two siblings. My brother is Professor of Medicine at the University of Pittsburgh. My sister is a lawyer.

Just as an aside, my brother and I were offered faculty positions at Stanford in the same week. It was my second academic position, his first. He’s two years younger. But also because he has more schooling, he was later in entering the academic ranks, just because graduate school takes so much longer when you do two degrees. So we were both applying to a number of places. I remember when I got the offer from Stanford, I called my parents and said, “Well, should I call my brother?” And they said, “Yes.” It turned out he had heard three days earlier and done the same thing. Called our parents and said, “Well, should I call Eric?” They said, “Well, why don’t you wait. He’ll certainly call us.” In Computer Science terms, we had a data-flow graph and my parents were waiting for both tokens before it would fire.

**B:** So the two of you, the two boys, followed scientific-type paths and your sister different …

**E:** Certainly, my brother, as a physician, has a scientific bent. But his area is in decision making and economics. His undergraduate degree is in economics, so he has social science bent like my father.

**[04:37]**

**B:** Can you tell me a bit about your schooling?

**E:** I went to public school in Nevada. All three of us did; the same school. And my brother and I both went to Harvard as undergraduates.

**B:** We’re going to stay down in that elementary school. Think back on that.
E: Think back.

B: Think back on that. And think about what characterized the school. What characterized the learning environment? Were there teachers who were particularly influential? What was it like?

E: Well, it was a very small elementary school. This was when I was living in Reno. Reno wasn’t very large in the late 1950s. I don’t remember my elementary school that well. I think class sizes were around 20. I was accelerated. I skipped the second grade. And I think one of the nice things about a smaller school system is that they take more time to make exceptions, to find ways of (allowing children to take) courses in advance of their grade level. I took my high school algebra course in 7th grade and was (well ahead) in mathematics. I had the opportunity to do it there.

Probably the most interesting piece of my education before college was that (I had the chance to go to school abroad. My father taught in a number of countries), particularly in later years after I was already in or out of college. His area of interest was comparative public administration and so he would go to other countries, study — either working for USAID or other travel scholar programs — and teach for a time at a local university, but also study and then write papers about the governmental systems. In his doctoral dissertation some years before, for example, he compared the bureaucracy in Sweden with the United States. The most interesting thing was to have the chance to live in Lahore, Pakistan, for two years as a child.

[07:22]

B: Could you tell me the name of the town, please?

E: Lahore.

B: Lahore.

E: The second largest city; the capital of the Punjab, very close to the Indian border. This was before the war in 1965, and actually the two countries were at a level of relative calm that has only been re-created sporadically, very rarely since that war. It was possible to drive across the border. There were, of course, border patrols. We were able to visit Delhi and Agra and Kashmir by car. It’s very difficult to do that now.

B: How old were you?

E: 10 to 12, so I remember it quite well. And remember the things that you learn when you’re 10 to 12, you know. So people will ask me things about the politics or the way people lived, and I don’t know the answers to those things except as I’ve learned them through other readings. But certainly the images of the country and of the town, and of visiting all those places, going up to the Khyber Pass in Afghanistan. I know — particularly given that these places are in the news so much now — to have been in most of them is interesting from a different perspective. But I think, more than that, the opportunity to live outside the United
States gives you a very different perspective on the world and how the US fits into it. We went to an American school when we were there, although there were Canadians and British and Pakistani students in the school as well. But we were living in a different country and a country that was undergoing turmoil. And there was some anti-Americanism at the time. And I remember all that. So it was interesting.

**B:** Can you tell me something about what you remember about your math and science education in school, was it … ?

**E:** As I said, I took …

**B:** Accelerated.

**E:** Accelerated. I took the math courses as soon as they would let me and always really enjoyed it. I knew I wanted to be a scientist from a very early age. In fact, I wanted to be a professor of science … when other kids wanted to be a fireman, I wanted to be a science professor. I didn’t quite know what that was, but living in an academic household I knew what a professor’s lifestyle seemed to be and that was very attractive. My father was home a lot, if nothing else, and working at home and reading was so important. I also went to school every summer that I can remember. Partly in demonstration classes for the education school at the University of Nevada, but then there were summer programs for gifted and talented youth, I think they’re called these days. And that’s what I was immersed all my life, in sort of, in the educational process.

[11:00]

**B:** Can you tell me a little bit more about some of the summer programs?

**E:** There was this marvelous program, and I don’t know exactly how it was constructed, but the University of Nevada had someone from the School of Education teaching philosophy. I remember that I read Plato’s *Republic* in that class. I couldn’t have been … well, since it was before we went to Pakistan, I could be at most in the 5th grade. And I don’t remember which summer it was. But there were students in that class of all ages in pre-college time, so it was an odd mixing. But it was fascinating to hear about all these ideas. To read plays. To read classical philosophy. To read science. I don’t have a good memory of exactly how this came up, but I remember there was an article in the Reno paper about the class and I was listed as reading Einstein. I don’t remember reading Einstein, but knowing that I wanted to be a scientist, it does not surprise me that I was reading something about Einstein.

So, I took those courses. I took chemistry a year early. I took physics. And in the sort of Sputnik-era attempts, particularly in the 1960’s when the Kennedy and then the Johnson administration starting pouring more money into schools across the country, to improve their science education, we had an electronics class — which I think was intended to be vocational — but the physics teacher took on electronics class I and II. And in the second year, there were only three of us in the class, so we had tremendous help from the teacher and it was amazing. I remember learning a great deal about that. I went on … well, we’re getting slightly ahead, so let me back up a little bit.
Getting into computing at a very early age was something that came out of the academic background. In the summer after we came home from Pakistan, so that would be the summer of 1965 or 1964, we came home that year — my father decided that since I was a science type and he knew, at the University of Nevada, people in the Engineering School, that he convinced some people in electrical engineering to take me on as a summer intern. And I was soldering wires on the back pane of an analog computer in 1964. You know, perfect for me. I could then and probably could now make very good solder joints. I had small hands so I can do it very accurately. I had no idea, I think, of the big picture. It was called a computer, so I was working with a computer. But I don’t remember really having a sense of how it worked. It wasn’t a modern, digital computer. But I loved the electricity in it. I had invented, as an 8 or 9 year old, the concept of a relay, an electromagnetic switch. I needed it for something and never had heard of one, but could have been able to do some digital logic at that time if I had just pursued it a little further. And learned years later what it was that I had done was sort of interesting.

When I started high school, my parents decided, partly for the school systems, to move 30 miles south of Reno to Carson City, which is the state capital. They thought the schools were better. I understand retrospectively that the Reno school system just said, “Oh, you should go to the University of Nevada as a 12-year-old.” And my parents didn’t want that. They worked out with the Carson City school system that there would be ways of getting me into the science classes I wanted a little early and giving me some individual attention there. I didn’t know about most of these things at the time, but it worked out really well.

And what my father did, rather than commute to the university in those years was, because he studies public administration, he got a job as the Deputy Budget Director for the State of Nevada. So he could learn how the bureaucracy works, and then write about it and go back to teaching. So he lived in the state capital. And all of the state offices for Nevada at the time, I think, fit into two administrative buildings. So here was the Budget Office on the same floor as the fledgling Data Processing Office. This is in the summer of 1965. So he got to know those people.

And it’s just an amazing story. My father brought home — I remember doing this — an IBM programming aptitude test that they gave in the 1960s and he gave it to all of us. My sister was probably too young to have done it then, but to my brother and to me. And I evidently got the highest score they’d ever seen. And so the director of Data Processing, the state director, took a personal interest in me and had me come down and work on an IBM 1401 computer. I worked with a number of earlier data processing machines, I don’t think you can call them computers. I have programmed collators, the IBM collators that you did with plug boards and wires. The 1401 had no permanent storage. There was no disk attached to it. I don’t even know that there were tape drives. Everything that you did was read in from the card reader, processed in memory, and then printed on the printer or possibly punched. And all the coding was done in assembly language for the machine. And I loved that. I wrote a number of programs. I wrote programs that the state of Nevada used — after school and later summer internships.
So, here I’ve become quite proficient in early high school at using computers of that
generation. I didn’t think of it as science. It wasn’t. It was business data processing for the
state of Nevada. I did have the opportunity to work with those things and certainly to be in
the machine room and see how all these computers worked. And I just loved it.

And then in the summer of 1968, that would be the summer after my junior year, I went on a
National Science Foundation funded summer science training program. So many people in
my generation did precisely that. There was a lot of them nationally to try to get people who
were interested in science the opportunity to work on a college campus, to take some courses.
And I still thought that I wanted to be a chemist. I loved my chemistry teacher in high school.
He was so good as a role model.

[20:05]

B: Can you describe some of the characteristics that made him a good role model?

E: Yeah, I could do that. It’s all of a piece, you know, moving back and forth in time. I took a
chemistry class. He read stories to the class. He was funny. He was engaging. He cared a lot
about his students. Most students, for example, in that day, as seniors took only a half a day.
So small a number of students were going on to college and very few out of state. Some
would go to the University of Nevada, from that public high school. And most would get jobs
in the afternoon and that was the transition to the work world. And there wasn’t that much in
terms of offerings. The highest mathematics that was offered in that high school was a one-
semester trigonometry. There was no calculus. There were no AP classes. Quite a different
public school than it would be today.

But there was the opportunity to learn and the school gave me that. My job in the afternoon
was being a lab assistant for the science program. And what John Hunter, my chemistry
teacher, allowed me to do was, I taught second period chemistry. I gave every lecture that
one year. And it was amazing to have that opportunity. I was younger than a good fraction of
the kids who were in that class. But it certainly taught me about teaching. It taught me about
lesson planning. It taught me about how other students were thinking. And I had a mentor
who was really good at making sure that what I did worked. And I don’t think that the
students felt in any way cheated by it. I had good friends in that class. People thought I was a
little odd but, you know, it wasn’t a hostile environment, which I think at some times is for
students who see themselves as more academic or more advanced. So that was what
happened in high school.

The summer before I did that teaching was the summer I spent at the program in Chico,
Chico State College, in chemical equilibrium and computer science. When I tell people the
story about this, it’s sort of rehearsed in my head, so it’s easy to tell. I chose it because it was
chemistry. But the idea that there was a little computer science helped. So of all the summer
science training programs that one could apply to, that one stuck out as a good opportunity.
And the schedule for that summer was every day we had chemistry lecture from 7:30 to 9:00.
People think nowadays, “How could there be something at 7:30am, students are never up!”
But in Chico, California, you had to do everything in the morning because after 2:00[pm], it
was 110 [degrees Fahrenheit] in the shade and you didn’t want to be in a classroom then.
So 7:30 to 9:00 was chemistry lecture, which I invariably attended. And 9:00[am] to noon was chemistry lab, which by the second week, I stopped attending for reasons that will become clear. Lunch, then 1:00[pm] to 2:00[pm], I think. Just an hour of computer science class, which was a programming class in Fortran on a 1620 — a really modern machine, which I just loved. And then every single day for eight weeks from 2:00 in the afternoon until 7:30 the next morning, I lived in the computer room, programming. And the only time I slept was during the chemistry lab. The directors called my parents to say, “Is this normal?” (I didn’t know that).

But I was doing interesting and exciting things. I was so obviously taken with it that about week four the computer science teacher said, “Why don’t you go work?” And he assigned to a group of graduate students doing a research project. I had no idea quite what I was … and I helped them code. It was unbelievably exciting, engaging, and so concentrated that I couldn’t have done it any other way and learned as much as I did in that time with that obsessive-compulsive streak that I’ve always had. By the time I was done with that I was just an ace coder in Fortran and assembly language. I coded in assembly language for the 1620.

The first course I took when I got to Harvard was …

B: I want you to step back a bit. So step back and tell me how you chose Harvard. What was your decision-making process?

E: Process? I applied to a number of schools. I wanted actually to go to Swarthmore. The Quaker connection was important to me. These were the years of the Vietnam War. I was active in the Quaker meeting in Reno and I even … We lived in Carson City but I drove, as soon as I could drive, each Sunday. My grandmother was living in Carson City by that time and was still that role model for me, as someone who put the health of the world above other things that she did.

My exposure to the larger world, outside of Nevada, I think came as much as anything through the American Friends Service committee, which would run these large high school conferences in California for people from the western states. So I was exposed to a lot of things that were going on. There was a conference that I attended a couple of years. The one that was so memorable was the one that happened right in 1968, one week after Martin Luther King was shot, and being together with other Quaker and non-Quaker high school students from all over.

I don’t believe that we had a single African-American student in our high school. But in Los Angeles there were plenty and those students came. And so I was able to work more with students from many different backgrounds and really get a sense of the world through that. And had done some draft counselling through the [American Friends] Service Committee in Nevada.

And so I thought that Swarthmore would be the right place. I didn’t get in to Swarthmore, but I got in the other places I applied. And the final decision … Retrospectively, I’m happy with that. Thirty-five years later when I was appointed the Lang Visiting Professor for Social
Change at Swarthmore for a year, the Provost called me to tell me that I’d gotten that position. I said, “It’s sort of fun, I didn’t get in to Swarthmore.” He was silent for a few seconds. “Swarthmore was the only school I didn’t get into.” [laughs] We had a good laugh about that. But the final decision was between MIT and Harvard. My father was a Harvard College graduate too, so that may have had something to do with it.

B: What about your mother?

E: Mother has a Master’s degree from the University of North Carolina.

B: And her undergraduate?

E: Greensboro College. My mother didn’t work professionally during the time that she was raising us, as I think most women in that generation didn’t do. She was comfortable in an academic household and read a lot, but did not, I think, sadly, pursue that intellectual career.

She’s very proud. My parents just celebrated their 60th wedding anniversary. They’re still together, still alive. Failing health, as one would expect at that age. But they are so proud of the three children that they’ve raised, all of whom have been extremely successful professionally. So I think Mom lived in some sense vicariously through the children in that, certainly followed the pattern that so many of her contemporaries followed.

[30:27]

B: So between Harvard … you were trying to tell me the difference … how you chose Harvard over MIT.

E: I don’t think the fact that my father was an alumnus made that much of a difference. What really made the difference was I knew that I had interests that transcended the sciences. And I recommend this to students today, that if they know that they are really interested in science and technology, and that they are not really concerned with how good the Shakespeare course is, you probably can’t get a much better scientific, technical education than at MIT in the United States. It’s just an amazing place. But that, if they want something that’s more balanced, the Harvards and the Stanfords of the world have something really important to offer there.

And I think that was true for me. I was interested in a number of other things and I enjoyed the fact, for example, that I could take three courses in folklore when I was an undergraduate and just loved them. I’ve always liked reading, and I’ve liked literature. I’m married to a poet. When I say I read three books a week, they are not computer science books. They wouldn’t even count in my list. These are novels or histories or something of that sort because that’s the kind of thing that really keeps me plugged in to the world.

It’s what I think … progress is for, to give people more of a time to be creative in that way. And I like computing because I think it’s creative. I think that the skills that I have and that I use are the ones that … they call on the same parts of the brain that other creative people have. I think that when you look at the kind of things that really excite people about any discipline, they have to have that creative strength; Paul Graham’s book *Hackers and*
Painters, it’s an important thing. I think that’s how I feel too, that my work is as painting is
to painters in some way. So, I decided very much …

Oh, the other thing is that given that I was deciding on colleges in 1969, the politics of a
place like Columbia or Harvard would have made those more exciting to me as a many
generation left-wing family upbringing. So I wanted to be at a place where students would
continue to oppose that war that was going on. One certainly heard more about that at
Harvard than one heard about that at MIT. So that probably had its influence. And the
Quaker Meeting House is right across the street from Harvard and just down the way. So that
was important too.

[B: Tell me a bit about that undergraduate experience …]

E: Worked out?

B: Yeah.

E: Well, it was an interesting time. I came to Harvard intending to major in chemistry and
physics. It was a joint major. I didn’t. I was terribly behind so many of my fellow students
because of the fact that my high school had not had Advanced Placement offerings. I was
placed into the advanced beginning calculus, it was Math 11, at Harvard on the strength of
my mathematics test scores. But almost everyone in that class had had calculus before and
was using it to review … perhaps a spotty high school education. And for me it was all new.
And one of my Stanford colleagues — one of my obvious contemporaries, since we were in
that same class — and I talked about that class in years later. He loved it; I just … didn’t.
And he became a mathematician and I became a computer scientist. I loved my chemistry
teacher. But in the final analysis I ended up spending — in more or less the same obsessive
way I had in that summer program — ALL my time working in computing.

In my sophomore year, I had a chance to lead some review sections and later to teach a
section of the introductory computing course — that’s still something our Stanford students
do, that those undergraduates do. John Hennessy, President of Stanford University, said once
that he doesn’t think he would have gone into academia if he had not had the chance to teach
as an undergraduate at Villanova. That that just opened his eyes to what the academic
mission could be. And of course, since I came from an academic family I had more of a sense
of that. But the chance to do again what I’d done in high school teaching a class as an
undergraduate made an enormous difference. And that was possible only in computer
science.

I loved the computer science teachers that I knew at Harvard. They were really genuinely
interested in taking their passion for the field — not the whole department, but the people
that taught the introductory course. One of them was Professor of Biology named William
Bossert. He taught the introductory course for years and years and years.

B: Bossert?
E: Bossert. He’s still teaching as far as I know. Saw him just a year or so ago. And he had his community: the students that he taught, the students who taught for him. So many of them have gone on to academic positions because of that example that he provided … in much the same way that Brown students have felt about Andy van Dam over the years.

B: Did they have a program in computing?

E: No. It was called Applied Mathematics until, I think, 1989; I could be off by one or two years in the history of when Harvard actually named their program. I got three degrees from Harvard — bachelor’s degree, Master’s degree, and a Ph.D. — all of which are in Applied Mathematics. There is more mathematics than there would have been in a computer science degree of the same time, which I don’t regret. But it was, for all intents and purposes, a computer science degree. It was just … it was too new a field to … I remember talking to a Dean at one point when I was a graduate student who still believed that computer science was just a new and presumably flash-in-the-pan fad that would soon pass. And Harvard needed to defend the long-term ideas. And that we should wait until this matures in a generation or two and then maybe we could name it as a field of study. Which eventually, of course, they did.

B: So, in your mind, do the degree paths, the Master’s and the Ph.D., do they meld into one seamless whole?

E: Oh yes. I had decided to do computer science — if not in name, in fact — by the end of my sophomore year. And that was the only major I declared.

B: I think what I am trying to say — the educational experience. Do you differentiate your undergraduate program from …

E: Not much, not much.

B: … your Master’s and how did that

E: Well I was at the same institution which of course I …

B: But you chose that.

E: Right, I chose that. I subsequently counselled students not to do what I in fact did, as many of my faculty members counselled me to go somewhere else and have a different perspective. But I made that choice, not for academic reasons but for personal reasons. I wanted to stay in Cambridge, which I loved. I didn’t want to leave friends. I didn’t want to leave the woman with whom I had been involved for a number of undergraduate years, although that didn’t last until the present time. But you know, people make those decisions. It wasn’t a bad decision. It’s just that it wasn’t necessarily the most mind-expanding decision. And so the fact that things seemed to be a natural progression is probably easier to explain by having the same people, the same institution behind it.
B: Can you think of ... were there any women involved in the field?

E: Not as many. I know Lynn Stein from that time — sort of. At Olin now. She was singular. This was the year I came back to Harvard when I was teaching at Wellesley, so it was some years later. But it was so unusual to have women involved in computer science in those days. There were a couple of doctoral students who were female.

And what’s fascinating is that my life at that time was bifurcated. I had a political and social community that was almost entirely women. The political feminist community centered around Radcliffe and Cambridge. The Redstockings, sort of the second and third wave — depends on how you count it — feminism at that time, was very much centered in Cambridge. And those were my friends. And then there were the people that I studied with and worked with professionally, who were not by and large my friends. I wouldn’t see them outside of classes. And one was almost entirely male and one was almost entirely female.

I have very strong role models of women in my family. My grandmother, my father’s mother, for example, was probably the single most important influence — I’ve said that in other interviews — growing up. I dedicated my first book to her. And what she did in terms of making it clear that I could accomplish things and being very active as a feminist force in her own … in her way meant that I was really, I think, ready to go that way. But in part it was the accident of my living at the Radcliffe quad and finding the first … feeling at home there in a way that I had never felt on the Harvard side of things, partly for issues of class.

There’s this amazing historical fact that I arrived at Harvard before there were co-residential dormitories. Into an institution that was about — again I might be off by some small percentage — I think 55% private school graduates. During the time I was there, their co-residential experiment started. I moved to Radcliffe, right at the beginning of that and stayed there for many, many years because I lived there as a resident tutor when I was a graduate student. And so you opened the floodgates and women who really liked being at Harvard moved there and men who didn’t like Harvard, I think, moved to the Radcliffe quad. The Radcliffe quad was over 90% public school graduates within three years. The sorting was not done on anything other than sort of comfort level with the enormous class privilege of Harvard. Radcliffe is not a lower-class institution anywhere in the country except at Harvard.

And people who are frustrated by the class focus and the left, the people who have gone to public school, the people who felt out of place at Harvard, gravitated toward that island of sort of normalcy in the midst of this wildly skewed social environment that was Harvard. And what happened subsequently, and continued for so many years afterwards, was there was this attempt by Harvard to steamroll Radcliffe and just draw into its orbit everything that had been important historically for women students at that institution. And those of us who decided that we liked the things we had found at Radcliffe fought it. And my allies were that feminist community.

[45:19] And, I don’t know, that was the people that I interacted with. And so, when I got my Ph.D. and wanted to go find an academic position, I went extremely deliberately to Wellesley.

Again, it let me stay in the Boston area where all my friends were, or where that community
was. But, it was also the opportunity to create a program at a women’s college. I was the first computer scientist they hired. And in the course of the five years I spent there, a major was created. And I believe that you could detect, in just looking at employment statistics for Massachusetts in that sector, that the existence of a program at Wellesley made a difference in the gender make-up of computer science jobs that was measurable. I remember the statistic that was just so astonishing is that in 1983/1984, Mount Holyoke College graduated more women in mathematics than the entire Ivy League, even though it is smaller than every institution in the Ivy League. That’s what the women’s colleges did. They were able to produce people in fields that the stronger Ivy League and Little Three colleges [Williams, Amherst, and Wesleyan] never did. And I wanted to be part of that. And was part of that. And then decided for other reasons to go to the west coast.

B: You want to talk a little bit about that transition? Or … before you do that; talk about your teaching philosophy. Sounds like it started in high school and it was perpetuated …

E: I learned it at my father’s knee if you will. He’s also taught as I have — broadly. He uses many, many different resources in his classes and teaches courses where, in looking at public administration, they read plays. Because he thinks that, in fact, the best place to look at the way people treat each other is not in … of course he assigns analytical texts by people who do public administration or political theory or sociology; those are part of the reading. But to really have a sense of how someone responds to power without — his course on plays reads Antigone, either in Sophocles’ or Anouilh’s version, and Ibsen’s Enemy of the People, and Bertolt Brecht’s Galileo, to look at the question of the relative responsibility of the individual and the state — and Antigone is probably just a classic in that. And so I remember in high school reading those books when he would assign them because I’m fascinated by that and he would talk about them at home. And so I think that the notion that a broadly-based education is essential pre-dates even my teaching in high school by a lot. I grew up with that.

B: So can you give an example of what a class was like or some of your classes at Wellesley?

E: Wellesley. Well, Wellesley was complicated in that I ended up leaving because, you know, I couldn’t, in some sense, make it work and make it survivable. There was a phenomenal growth of interest and enthusiasm in computer science in the early 1980s, you probably … you know all about that. But I got my position at Wellesley in 1980 and over the next 4 years, the doubling rate of students — it was faster than Moore’s law, doubles every year — and there were no faculty to hire. It’s 1983-1984 that the ACM estimated there was one applicant for every seven faculty positions in the United States in computer science. And so all these schools are trying to handle increasing numbers of majors in an environment where they cannot hire additional faculty, there just are no people to hire. And so they retreated mathematicians and physicists and bring them in to teach computing. All of that was the usual approach.

[50:38] Wellesley was a little bit more fortunate because we had the attractions of being one of the top colleges in the United States, but also in the Boston area, which a place that people like to
live. So we did get applicants in that year and in previous years and were able to build up our program. But the teaching load was still ... I wrote a memorandum where I was able, I think, to prove quantitatively that the load on computer science faculty was more than twice our nearest rival and almost four times the average. You know, in the number of students we taught. The student classes were supposed to be capped at Wellesley at 35. I routinely had more than 70. And no graders. But how could you do anything else? Those students wanted to take it. I mean, I could have taught five courses in a quarter instead of three and reduced the number in each class, but it doesn't change the work load in any way. And the demand was there. We couldn't satisfy the demand and couldn't bring in enough people.

We did bring in Jim Hendler at Maryland, who started his first teaching with us. Evelynn Hammonds, now the Dean of Harvard College, I hired to teach computing at Wellesley. So there were a number of people that we did bring in, but we couldn't, in that, reduce workload, not during that, no question. I went back and taught in my year of junior leave at Harvard. I went from teaching three classes in a semester (I said quarter earlier because that's so ingrained in my Stanford world) that I would teach three courses and then I went and taught one course at Harvard.

The courses that I taught at Wellesley were entirely female. The course I taught at Harvard had one woman. That's when I got to know Lynn. She was so easy to spot by being all by herself as a woman going into computer science at Harvard, intending to make it her academic career. And so we had ... it was very rare. And I wanted to change that. I wanted to change that when I saw that all of my friends — albeit political friends, who were, as I say, almost entirely women — had real trouble finding lucrative jobs in the majors they had chosen because they were passionate about them. And all the men that I knew as colleagues were going off and getting these great jobs in the Route 128 build-up in the 1980s, or late 1970s or early 1980s. And so I wanted to see what I could do to make some difference there. And that's why I sought jobs only at women's colleges.

B: So after your junior sabbatical at Harvard ...

E: I didn't go back.

B: You didn't go back. And were there other places you were looking other than Stanford?

E: Well, I didn't go directly to Stanford.

B: Okay, where did you go?

E: I went to a research lab for five years.

B: Right ...

E: And ... it was the lab that was formed at Digital Equipment Company out of the collapse in 1985, 1984-1985, of the computer science lab at Xerox PARC. Xerox PARC, of course, had invented all the technology and refined the technology to some extent. The genesis that's in
that milieu of PARC and Stanford Research Institute, now SRI. The computer science
department at Stanford. All the ideas for bitmap displays and the mouse and the user
interface technology that we consider just standard today grew out of that time and was made
real at Xerox PARC. But it wasn’t marketed by Xerox. This pair of entrepreneurs named
Steve Jobs and Steve Wozniak in their garage came and looked at that and hired a number of
people away from Xerox. And the Apple Lisa and Macintosh were that work that had
happened.

[55:49]

I don’t know if you know the book by Alexander and Smith called *Fumbling the Future:*
*How Xerox Invented, then Ignored the First Personal Computer.* A very interesting historical
time and interesting too — I know economists and business leaders who think that Xerox did
exactly the right thing. They didn’t commercialize that technology successfully. They tried to
market the Xerox Star, which was the commercial version of the Alto. And it didn’t work;
the price point was completely wrong. No one thought that this was going to be on
individuals’ desks. Rather, this would be a workstation that would be used in a company.
And so the whole PC revolution missed. The timing was just wrong for Xerox to
commercialize it. But Apple succeeded.

And so the powers that be at Xerox got very upset that they hadn’t been the ones to bring this
technology forward. And Bob Taylor, the director of the lab, was fired and enjoined not to
talk about his firing. He violated those terms and managed to take almost the entire technical
staff with him. And knowing how good this group was, Digital Equipment Company set up a
lab in Palo Alto — two actually — hired most of the old team. Taylor directed it and hired a
number of people, many of whom were people I had known as an undergraduate at Harvard.

And there was this wild day that I was visiting the new lab. And in the course of visiting one
of my good friends there, I ran into six or seven other people with whom I had a strong
connection, most of whom had lived in the same dormitory (the same dormitory that Bill
Gates lived in in those days). It was old home week! And the possibility of doing good,
collaborative research again got me so excited that I abandoned teaching — which looked as
if … 24/7 was not enough time to make that work. And I didn’t see how to expand beyond
the 24/7 I seemed to be working. And so I thought maybe working a mere 80-90 hours a
week I could do things that would matter and it would be fun. So I applied for and got a
position at that lab. Only to discover that I missed teaching too much. And then five years
later took the position at Stanford.

**B:** Well, clearly you applied for that position because you and your brother were both
applying to Stanford at the same time.

**E:** Yep, yep.

**B:** And that was the only teaching position that you applied for?

**E:** No, actually that year I applied to Mills College as well and I got both positions. And … it
was touch and go. Now, Lauren and I were together by then and had been for a number of
years. She would be a good person to talk to about my agonizing over that decision. I usually
tell people that had the year after … my first year at Stanford was the year that the Board of Trustees at Mills College decided that they would make the school co-ed. And the students at Mills successfully organized against that decision. Had that been a year earlier, I probably would have gone to Mills because I would have been so impressed by the dedication of the students.

[59:54]

But I felt that I had more of an opportunity to make a difference at Stanford, that the things that I would do in terms of education would be seen by more people. I really liked the idea of having very, very bright students. It’s not that the students at Wellesley and elsewhere were not bright, but they weren’t focused on technology in the same way. That became very clear at Wellesley. And I tried to do things to change that orientation, to develop the kind of passion (if you’re being generous, obsession if you’ve decided that you don’t like it — but two different sides of the same coin). That there weren’t people who spent their lives doing this work when I taught at Wellesley, and that I didn’t think there’d be at Mills. As there had been at Harvard and as I knew there would be at Stanford.

B: I want to step back just a bit because I want you to inform me and the listeners about the research path that you took both as Master’s level, Ph.D. level, that has to have carried that passion. Or did it carry the passion?

E: Probably not.

B: Probably true. Because you talk about your passion for life came from the women.

E: My passion for teaching is different from my passion for research. I don’t believe that … I mean, I’ve never believed that the best researchers in any sense make the best teachers. They may, but I don’t think that they’re correlated. One of the myths that I think the research universities promulgate is that the great thing about being at a Stanford or a Harvard is that you get to work with people who are at the tops of their field. It tends not to be true. A few people do, but only the ones who have self-selected and so are sort of brought into those research groups. More now at Stanford than was true in that day, I think that that myth is becoming less of a myth and more real than it was because of continual reform. But that what has always been true in computer science at most institutions is that some people have been brought in whose passion is in teaching. Because of … a variety of, sort of, demographic factors. We have … I mean, for the last 20 years, more student units are taught in computer science at Stanford than in any other department. And if you ask my colleagues that across the university faculty, no more than 5% could tell you that. It’s one of the secrets.

B: Describe the term “student unit”.

E: It may not be universally applied, but if you take each registration and the number of units / hours / credit hours that are assigned in that course and you add them all up, we’re probably 50% more than our next closest rival, which is electrical engineering. Stanford is, of course, an engineering … is more of an engineering school. I don’t believe that this is true at Harvard; I don’t know the numbers. And because we have an enormous graduate student load, particularly in the Master’s program, people taking courses. We have one of the largest
majors, undergraduate majors. Those give us, in some sense, the same load that other large
departments would have.

But what we have that no one else has is that everybody takes the Intro course. And maybe a
lot of other people take courses for quite some time. Our service load is huge. When I was in
the Dean’s office at Stanford, I looked at these numbers. And if you look at what every
engineering department teaches in terms of what fraction of their units … again, you have to
scale it somehow because at Stanford, a course is not a course. There are some courses that,
you know, are one unit big and they meet for one hour a week. And you can’t give that the
same weight as one that subsumes its students. So when you scale it by the number of credit
hours, that … in our department, we teach less than 20% of our units to our own students.
Four out of five or more go to students in other disciplines who are taking this course. So, we
brought in lecturers, teaching faculty to — and using undergraduate teaching assistants — to
meet that demand. Our faculty still teach more student units per faculty member than the
average in the School of Engineering, even though we have this huge para-professional
teaching faculty along the side. It’s not that they’re not doing their job, which occasionally
we hear (“Because look, you’ve got these lecturers!”). Yes! But our faculty are working
harder doing their job by any number you can count. So, that all comes from the enormous
demand from outside the program.

B: I guess the question I have as you look at that is: At Stanford, is there a separate
teaching track?

E: Yes.

B: Tenure track teaching …

E: No, it’s not a tenure track.

B: It’s not a tenure track.

E: It’s a permanent track, but it doesn’t formally have tenure.

B: Do you have labels? At the professorial levels are the same.

E: Yes, absolutely. I’m a Full Professor (Teaching). And I don’t in some sense have tenure. I’ve
had an endowed chair. I’ve been president of the faculty senate. I don’t ever get re-appointed.
I have a continuing appointment. But they could eliminate the program and they would not
feel obligated, necessarily, to find me a job teaching in some other department. I’ve never
thought that this mattered one whit [laughs]. I don’t believe it does.

And we have … we’ve hired a second person in that professorial track and made an offer that
may still get taken up for a third. It’s … and all over the university there are people in that
line. And they’re the most successful teachers, in general. And they’re long-term contracts. I
mean, I expect to retire there. I guess I can’t imagine the programmatic need would ever go
away. And as I say, it’s not as if one is continually re-appointed. In fact, my appointment
after the first re-appointment decision when I was re-appointed as a full professor. Yepp. Nobody has paid any attention to me in terms of re-appointment. I get good salary increases because they think I’m doing a good job, but it’s not … there is no sense in which a position is any less secure than someone who formally has tenure. At Berkeley …

B: The first part of the question … Yes, tell me about Berkeley.

E: At Berkeley, the lecturer’s have tenure. They are hired under …

B: That’s part of the State.

E: Yes, that’s part of the State system. They have the same security of employment the faculty do. I think it’s called something different, but it’s effectively a tenure decision. And sometimes it makes it harder for us to recruit since our terminology is different. And yes, there is that possibility that the university could decide the programmatic need had gone away. But I don’t believe it’s ever happened so, you know, I’ve always believed the best defense against being laid off is being indispensable [laughs]. It seems to work.

B: You are concerned about computer education and concern about women in computing has permeated …

E: My work over the years.

B: Your work.

E: And because, since I got to Stanford, what I’ve done as my work is computer science education, I didn’t really pursue … I mean, when I worked at the research lab, I continued my work in multi-processor systems. That’s what I did my doctoral dissertation on. I like it, but I haven’t done much with that since I came to Stanford, maybe nothing. So I have a reasonable track record of papers in that area, particularly in terms of its length. I wrote papers in the very early days of multi-processor design that were in the classic books about multi-processors that created the field. They were printed there. But … so I had a real contribution in that research area.

[70:01]

But it wasn’t where my passion was. My passion was for teaching computing and in particular teaching programming. And getting people to understand the amazing excitement that attends to the ability to be able to solve problems by computer. You talk to other people and this is commonplace, that everyone was captivated by the power that programming gave. And I believe it. I mean, I still believe it. I still spend … probably too much of my time coding. But I don’t know that I could do anything else. It is just the most fun thing to do.

You know, I was telling one of my colleagues that when I write my Christmas letter for this year it will almost certainly not include the thing that I found most exciting, which was working out how to do something in a class that I had convinced myself for two years was impossible to do technically. And then one morning, in the shower, you know, this epiphany happened: “AH! I could do it that way!” And you run and you make it work and somehow,
you know, that just keeps you going for months. I was very happy with that. But no one else
really outside the field — or really outside the people who are teaching that material —
would know how important it was to be able to restore the ability to teach in a way that we
had been able to do when were teaching in C into our new courses was just amazingly
important. I think I’ve argued in the past that the ability to build iterators in that way saves a
lecture day. So in a quarter system that’s probably 2% to 4% of the course, somewhere in
that range. That’s a huge savings.

B: Is your teaching primarily in the beginning classes? What courses do you teach?

E: I teach a wide variety of courses. I do teach the beginning classes. And the textbooks that we
use at Stanford and are used many other places are textbooks I’ve written. And I’ve
reconstructed that course over and over again. It doesn’t tend to stay put very long. You have
to track languages that become the most important languages to teach because those are the
only languages that you can find the right technical support for … the best compilers. You
need to teach on something that people can buy the software for. You can’t necessarily outfit
labs that will work with a thousand students a year, that’s just too expensive for one course.
So you need the things that people can just go out and download free compilers for. So that’s
got to be the stuff that is in the most common programming languages. And over time we’ve
had to retool that course and rewrite the textbook. And I’ve done that a number of times. So
that’s my primary work in the computer science department.

I think I teach almost every year the course in computers, ethics, and public policy — which,
of course, all computer science departments have something like that to meet the requirement
on social and professional issues because the curriculum guidelines or accreditation requires
it. So in computing, those are the courses I’ve taught more or less every year.

But I’ve also taught … I believe it’s now in 12 programs and departments at Stanford. And in
terms of my courses are at least cross-listed in … psychology was one of the ones I picked up
this quarter. And linguistics and philosophy and English are all included in that list of courses
where I’ve taught with colleagues in those departments to teach a combined class that looks
at issues from our different perspectives. And I’m certain that that’s a record for the school,
for the university historically. I love doing that. I won’t be doing it this coming year, but for
the last three years I’ve been on the faculty in the program in the introduction to the
humanities at Stanford because I want to model, as a real person, a technical person, an
engineer, a computer scientist who thinks it’s really important for students to learn how to
read. And I don’t mean how to read on the Kindle or read technical papers (those are
important too), but to read the literature of humanity because it will — not because I think
that that will make them better humans, I know that’s true too — because it will make them
better scientists, it will make them better engineers. I don’t think that without an
understanding of humans, you do a good job with that. And I, like my father, believe that you
can get the best insights into that, not from … there’s no way you can get from reading a manual
—a computer manual — much sense of how people think about things or how people learn
about them. And even the work that’s done in human factors and psychology and the design
division of mechanical engineering don’t really speak to the way people think about things in
the way that reading the best literature would allow you to understand. I think you can get an enormous amount out of that.

[76:36]

As an example: When I taught the course in introduction into the humanities, which is a jointly taught program at Stanford. All those courses are taught by a team of faculty. And the rule is that none of the faculty members can come from the same department. The idea is that it is not an introduction to a particular discipline, but rather multiple disciplines that look at classical literature. You choose a set of texts that go along with your course, that has historical sweep, that has some diversity of authorship so it’s not all some white male canon from classical times. You want modern pieces, you want pieces that are written by women and minorities. The course that I taught, with — mostly with a professor in the German department, but also with a professor in the English department one year — is a course called Technological Visions of Utopia, in which the question … the overarching question that we are asking is, “How do people perceive in literature writ large the changes in our lives that technology promises or delivers?” And so in the early years of Utopian writing, Plato and More, who are sort of the classical Utopists, don’t really talk about science and technology at all. And then you recognize it’s easier to find in Plato than it is in More. But by the time you get to Bacon and his New Atlantis, technology and science are going to liberate humanity.

But you get into the 20th Century and writers start thinking about technology as being much more dark. For instance, among the readings that we’ve taken during those — it’s changed a fair amount over the years that we’ve done this — Metropolis and R.U.R. [Rossum’s Universal Robots] as a sort of base. The Time Machine, the late 19th Century, it isn’t 20th, but the same idea of the possibility that technology might even bifurcate the species. The mid-century dystopias of Brave New World and 1984, both of which have technology, or at least biology in science, being critical components of their social control that each society puts in place. Modern cyberpunk fiction, when you read Neuromancer, for example, this is a dark world. It happens to have a particular fascination for a lot of people that are technology-focused, but no one could really like that world.

[80:11]

So what is technology holding out for us? The single reading that I think my students have found most compelling is the 1909 short story by E. M. Forester called The Machine Stops. It is absolutely wonderful. It is more prophetic. You know, last year was its centennial. And so when we are reading it we could point that out a hundred years before the present day this very accomplished English writer, who would provide the Merchant / Ivory film series with most of their stories, had a prophetic vision of a world in which we live in a room in which we communicate on screens and all our lives are mediated through the machine. It is probably far more accurate than anything else written before 1975. It is just amazing how closely he managed to nail it.

B: I’m going to stop that part of the conversation, just to ask that …

E: But those are the courses that I teach.

B: … to merge this. To merge this with your passion for programming and the importance that you think an education in programming and a love for programming is within a
computer science context. Do you teach any programming in those classes that are
cross-disciplinary and, if so, how does it fit? And perhaps what languages do you use?

E: Well, that’s an interesting question. One year we did sort of teach a little tiny bit of
programming in that course. It’s not … I mean, it’s a humanities course. It’s an introduction.
That’s what it says. So I felt it would be inappropriate, unless you could teach it as a
humanities discipline. But there was one year that our reading list included Neal
Stephenson’s *Diamond Age*. Neal Stephenson is a modern, very creative cyberpunk author.
We’ve tended to use his more famous *Snow Crash* novel in most years. The trade-off is that
*Snow Crash* is certainly more popular with students, and we can say a lot about it. There’s no
shortage of interesting points to make in terms of the course history. *Diamond Age* is less fun
for most readers, but certainly closer to the theme of the course. There’s a lot more to say,
and it has a female protagonist. *Snow Crash* arguably has too, but the person whose name is
Protagonist, and therefore you couldn’t possibly miss who he is supposed to be, Hero
Protagonist is the name of the hero. And protagonist of the story is male. So particularly if
we are trying to encourage more women to think about this field, *Diamond Age* is a little
better. But that story, the second half of it, is in the land of Turing. And knowing about
Turing machines helps you understand the novel. So I talked in that year about Turing
machines and had people program some. That’s programming — it really is — but it was in
the service of learning to read that particular novel. And it was a way that I could bring my
expertise to bear.

B: What percentage of your teaching is in the humanities courses then?

E: I teach about one course a year in that area out of four.

B: Can you talk a bit about … what’s a typical day for you?

E: I’ve always been an early riser, so I get up at whenever the sun does or sometimes earlier.
And I walk in to work. I do a lot of thinking while I walk in to work. One of the things I’ve
talked to students about a lot is that, because they never think about things when they’re
walking, because the cell phone comes out instantly. I have asked, “When do you, for
instance, if you’re giving a talk, when would you write that talk? If you’re writing a paper,
does it only get written when you’re focused on writing the paper?” Where’s the background
that, for me, is essential? All these ideas come out in the downtime when I’m able to focus on
something while I’m walking or in the shower or all those things when I’m disconnected
from the web, and not texting everybody every second. You know, that time doesn’t exist for
most students today, so I don’t know what replaces it. The evidence that we see from
sociology is that nothing replaces it and therefore they can’t do certain things as effectively.
But that time is important and I want to include it in my day.

[86:05]
I live a block and a half from the campus boundary, but Stanford being 11 square miles, the
trip to my office takes half an hour, all of it on campus essentially. And so I get a lot of
thinking done every day. And the same thing on the way back. I then write on … I guess I
spend, I don’t know … I would say that I spend three to four hours preparing each hour of
lecture, maybe more, maybe a little more. And almost all of that has to do with writing the
examples and the slides. It’s programming of a sort, you know, that I have to build
convincing … this is for the computer science courses.

I believe that the courses that I teach outside the field, even though I have to read whole
novels and read them every year — I actually believe that’s important, I don’t want to lecture
on some thing that I’ve only read two year ago, so I will and usually find new things in them.
Writing paper assignments in some of the courses, actually grading those papers, which
Stanford faculty rarely do. You know, “That’s the TA’s job,” but I’ve done it in many
courses routinely. Those courses — the courses that are non-programming courses — take, I
think, on average one-fifth the amount of time as a programming course does to put together.
I put a lot of time into them, but nowhere near what I put into the programming courses.
Writing an assignment, a new assignment, in the introductory programming courses, our
team estimates is an undertaking that’s somewhere between 50 and 200 hours. In other
words, a tenth of a person year. You couldn’t do that every year, you know, with every
assignment. There just isn’t enough clock time. So you have to reuse things and redevelop
them. And that’s unlike any other discipline that anyone can think of. Putting together an
assignment for the humanities courses or the ethics courses that I teach, I mean, I put some
time in, but nothing on that scale. Tons of it.

And so those are big pieces. So … but class preparation is a big part of what I do. University
administration has been a big part of what I’ve done for … that was … I worked in the
engineering Dean’s office. I directed the undergraduate program in computer science for
twelve years. I was chair of the faculty senate. I’ve chaired several of the senate committees
at one point in time or another. I was principal author of the student judicial charter that
replaced the old honor code charter a decade or so ago. Because, after all, computer science
has more honor code cases than any other department, I figured I should put my time in
there. So university administration takes a lot of time. Writing books takes a phenomenal
amount of time. It’s what I do mostly. It’s what I really think my job is, writing books. And
I’m working on books 6 and 7.

B: Titles?

E: Well, one is the *Programming Abstractions in C++* that I’ve just redone for the course that
I’m teaching. And the other … well, I haven’t … I mean, the book that I’m still working on
is the intellectual side of computer science, so I’ll probably be working on that for a number
of years. But I’ve … as I said when we were walking over, I really want to put together a
monograph, a short book, on the title *Why Programming Matters*, to talk about issues and
educational theory and the way one should look strategically at the computer science
curriculum and why the trends that we’ve been seeing in recent years to de-emphasize
programming are so damaging to the field. You know, that would be fun to do. So that
finding time, carving out time, which I use now the summers for, to do that intensive work as
necessary to write a book is critically important.

B: One of my favorite all-time computer science reads, and useful reads, was your
*Thinking Recursively.*
E: That was a good book.

B: And it was in Pascal, and it’s now in Java and somehow I missed that. So how do you think you can …

E: Well, you know, the Java book has not … I mean, it has not been successful. Neither of my Java books were successful in the way that their earlier counterparts were.

B: Why?

E: I don’t … I think I missed the moment. I don’t think that people are moving to Java at all. You know, there was this belief, that I bought in to, that Java was the best language for teaching introductory programming. I didn’t believe that from 1995, when it came out, until 2000. And we kept Stanford in C longer than most institutions did. But by that time I thought that there were good reasons to believe that moving to Java, about the time that the AP did, was the right choice. But it turned out to be so problematic that I think, by the time the book appeared, everyone who was going to change had already done so. And they had made their peace with whatever book that they were using. And so the idea of adopting a new book for a language which probably wasn’t going to be the lingua franca that we had hoped it would be — and that I wrote a paper, you know, “The Dream of a Common Language [: the search for simplicity and stability in computer science education],” taking the Adrienne Rich title and making that be, you know, what we might find, as we had in Pascal — that there might be a new language that would have that characteristic, and we thought it might be Java. It’s not. So that these books appearing in 2005 and 2007 didn’t really have that possibility. And then I don’t know that they’re as good a book, because when you’re working in a simpler language you can concentrate more on the big ideas and not so much on details. But I don’t know that, it also may be that books have not been as critical to the success of courses. I mean, there’s so much material available online.

I was talking a little bit with Rich Pattis recently about whether he and possibly I (and I would be happy to just give him all my stuff and have him do it), but put out a 30th anniversary edition, in 2011, of *Karel the Robot*. It should happen. And his argument is that, “Well, Eric your book is already available.” My book is available, my revised Karel, which we have a little copyright notice on front of, “Is available through resellers on the Internet.” It’s just there! You can get it, no problem, for people who don’t have any connection with Stanford. You know, it’s just … that, for books … the whole publications crisis is enormous. And you know, every year I get many complaints from parents who buy the 1994 C book from Addison-Wesley. And they complain about the quality: “It looks like it’s a mimeographed off-print comic sort of style that Addison-Wesley is selling for $95.00!” And I say, “Well, I don’t have anything to do with that. They’ve got to make whatever money they can because they’re not going to be in the business long if things continue to go as they are!” We’ll lose publishers, we’ll lose newspapers, and with it, the content creation that’s so critical. Blogs have their utility, but they’re mostly concentrators for people who have put money into generating that content. And when those content providers disappear, the bloggers won’t be able to replace it.
B: Can I move you to a different area now? Because we really have talked quite a bit …
There are two places that we haven’t mentioned. Have you ever … do you get grants and what do you … is that … you haven’t mentioned that as part of your …
E: I’ve gotten some. It hasn’t been that much of a career path. I’ve gotten, in fact, some very large grants. I had a grant of over a million dollars over the life of the grant to develop the computing curriculum in Bermuda. This was a grant from a consortium public/private partnership in Bermuda, so there was some money … mostly not applied to Stanford, because it’s difficult for a foreign government to justify, but that, in terms of the total expenditure, a lot of money was being spent in Bermuda by the government, but supplemented by the companies in Bermuda that needed to find a way of training people with IT skills on the island. Because the costs of relocating or bringing in people to Bermuda, given the enormous pressure against population growth on the island. The fact that you’re not allowed, for example, to bring in a car. You know, all those things make it harder to recruit off-island for that kind of person. And when, after years of Tory rule, the Labour party won the prime ministership in Bermuda, there was a huge increase in the incentive to provide indigenous black Bermudians — well, not really indigenous, I suppose, but the long-standing black Bermudian population — who are economically advantaged, you know, relative to the Caribbean, for example. Bermuda has, if not the highest, one of the five highest GDP per capita in the world and incomes are extraordinarily high. But they’re so much higher for white Bermuda and the British Bermudian legacy of colonization than they are for the population, long ago, of African origin, that there was a new political imperative to change that.

And so, when we were contacted by Bermuda Ministry of Education and asked whether we could develop a curriculum … I mean, everything … all the stars were aligned to make that work and my courses, or the courses that were developed as much by my students as me, have been taught now for a decade in Bermuda to every high school student. And it’s changed markedly the employment patterns on the island. And so we’ve used Karel the Robot there. We’ve used our Java technology there. We’ve built a really nice curriculum.

And, of course, there has been interest in that curriculum from all sorts of other places. Los Angeles Unified, the largest district in the country, has expressed interest, which we couldn’t possibly satisfy. Bermuda is … we put together a national curriculum that looks like an experimental curriculum anywhere else. It’s two high schools, that’s all there are. A manageable number of students. There are, I don’t know, 400 high schools in LA Unified? And the opportunity … the big cost … what we ended up spending most of our money on is not paying graduate students to design a curriculum — I mean, that’s cheap, relatively — to providing professional development opportunities for teachers so that they can teach the curriculum. That’s where the money goes in doing good pre-college education. And you have to over-prepare. You have to train more teachers than you need, because some of them are going to leave schools. They have the skills to go out and make much more money doing much less work. “Let’s do that!” So finding some way to make a self-renewing pool of teachers there. Each new generation is taught by the preceding one and making all that work.

B: It coincides with your passion, so therefore you do that.
E: It enables me to do things I couldn’t do if I didn’t get the grant …

B: I suppose that may be the answer to the next question I have, which is what professional organizations are you part of and how do those roles affect your career or your aspirations for the discipline?

E: Well, I’ve been heavily involved in a number of positions for the ACM. I was chair of the Ed Board for a couple of years. I was on the Board of SIGCSE, which sponsors this conference, for a number of years until I took on the Ed Board role. I was the principal author and mover-shaker for the Curriculum 2001 computer science volume. I did the Java Task Force leadership. So I’ve had a number of leadership roles in the professional society. Mostly because I think that those are important to disseminating and making it possible to have success with the materials that we have been able to generate. And if they are going become more widespread, they have to have some kind of backing. And professional societies are one possible approach. So are leading universities. (And I actually think that it might even be more successful if Stanford were to sort of endorse a curriculum. That’s just the way that these things work.)

But you need … you can’t be working alone. This may come out of my political history. Michael Harrington said, “An unorganized socialist is a contradiction in terms.” And so having been schooled at organizations through the political side, I think that we need to organize ourselves professionally, because the whole is greater than the sum of its parts. I was the president of Computer Professionals for Social Responsibility for six years and was on the Board for more than ten. I don’t do anything with them now, but was very active in the 1980s and 1990s. I’m on the Board of an organization that looks at … looks to develop the commitment that students have in the sciences and technology for using their work in ethical and socially productive ways. It’s … the organization is called Student Pugwash, named after Pugwash Conferences International, recipient of the 1995 Nobel Peace Prize. This group of senior scientists internationally that came together in 1957 around nuclear issues. That was spawned by a manifesto by Bertrand Russell and Albert Einstein. That organization is still looking to bring students together internationally to think about how science can be used in socially productive ways. So I’ve had a lot of involvement in such groups, which is probably why I tend to go home around 11:00[pm] after I get up at dawn.

B: It doesn’t sound like you need a whole lot of sleep.

E: I need more than I get, I’m sure.

B: Can you talk about any of the challenges … you’ve talked about a few: the challenge of too heavy a workload and not enough people to hire when you were at Wellesley. You’ve talked about some of the challenges a bit at Stanford, but not much. Have you had challenges that have made you rethink your career or that …?
E: Oh, absolutely! And I’m in the process of having more of them. I believe that the work of programming is getting substantially harder year by year. Not all of it. I believe it is far easier than it used to be to get 90% of the way to where you want to be and just wildly harder …

[105:19]

B: That 10%.

E: … to close that gap. Debugging is just qualitatively harder and different than it used to be. The kinds of problems I spend my debugging time on wouldn’t have existed 15 years, 20 years ago, and you get to these things which only come up when there’s the wrong kind of concurrency on network applications. And that when one person is using a browser and there’s this distributed system that involves many different layers. Almost every faculty member in computer science has talked about how their students, in the beginning, believe that the mistakes, the fault “dear Brutus” is not theirs. That whole question. You know, “The compiler is doing it wrong here!” And we carefully explain that, “No, it’s not the compiler that’s doing it wrong.” I believe that now 15% …

B: We used to say the hardware failed.

E: Right. Or something like that yes. But it can’t be in ourselves.

And I believe that approximately 15% of the problems that I’ve chased down now are bugs in code that comes from the manufacturer of … whatever it is that I’m using. Java is full of bugs. It’s gigantic. I don’t know that the bug rate is any larger than it used to be. But certainly the corpus is so huge that — and no one understands it all — and so you’re in this position … Here’s an interesting difference. I’ve talk about these things in code developed with someone who works at Google. The idea that when you’re debugging there are two seemingly related, but not necessarily distinct, activities. One is finding the bug and one is fixing the bug. And what all of us were trained to do is to find the bug and then fix it. But nowadays those are separable. I have many, many bugs that I can find, but not fix because I’m not allowed to reach inside where the bug is and fix it. The whole encapsulation process that object-oriented programming has led us to has, I think, made it much more difficult to fix the things that are wrong. You send an e-mail message to the … to Sun or whoever is the person in charge of it. And then three years later maybe it gets fixed. The time cycle is just not … so what you end up doing is coding around that problem.

But in some ways the other disconnect is even odder. That I find that there are many bugs that I can fix, but not find.

B: You don’t know why it happened.

E: Right, I have no idea …

B: But this makes it work.

E: … and found a work-around, I mean, whether it’s …
B: A good one!

E: … better than a window in the bathroom [referring to having earlier helped the interviewer fix a problem with the bathroom window in her hotel room], we don’t quite know what we’re doing here, but that seems to fix it. It is something that happens in software all the time. And there’s this … I think, a certain loss of satisfaction, or things seem a little less, you know, elegant, if you’re in that position as much of the time as I think we seem to be … I spend more time …

B: You spoke of the passion for programming, that that talk of passion seems to come from our generation. And I don’t hear the word love or passion used as much with the next generation. Are you attributing this in part to that …

E: Well, I think that’s because it is more frustrating than it used to be. When you discover that there’s a bug that “if only,” or a work-around, that’s more common. One of the things that happened in the Java Task Force was that we absolutely wanted to be able to determine the name of the main class or, equivalently, as it turned out, to be able to determine whether a class was loaded already. There’s a method that would give that to you. And if only the designers had decided to make it public, we would have access to it. It would save enormous amounts of coding. But in someone’s idea of what clients ought to be able to do, that was not part of it. And that kind of frustration … you know, I can see it, I can almost taste it — “Why didn’t you give me access to that?” — is a frustration that I don’t think we had in our day. As I said, when you’re coding at the client level of web applications and are writing JavaScript code, 90% of the code, by the estimates that I’ve seen, goes into type casing for browser incompatibility: “If this is Internet Explorer before version 4.0 do this. If it’s version 4.3 then do that. And then if it’s version 4.3 then do that. And then if it’s Netscape give up here! And if it’s Opera, you know, do something else.” That is not interesting and edifying programming. And it’s impossible to make it elegant. Moreover, it’s impossible to make it stable over any long period of time because new versions are being promulgated all the time and no one can keep up. So that’s a frustration. The lack of any convincing way to create a legacy in software is difficult. You know, that everything that you build is ephemeral and will be obsolete, if not on the day it’s delivered then some …

[111:40]

B: … day before …

E: Right, some time not so different from that, plus or minus one, right? And that I think people find, you know, difficult to really work hard if that is in your way. I mean, it’s very lucrative. But being in management is more lucrative, even companies where the rhetoric is that “code is king,” you know the Microsofts and the Googles of the world. It’s still the case that the management track earns more money faster, and people who may have liked the technical part at one time just are captivated by the pattern of advancement. Or the number of people from computer science and physics that prior to last year’s meltdown went into the financial services industry or working for money management or hedge funds or quantitative funds and helped precipitate the economic collapse that we are dealing with worldwide. All the incentives were wrong. You could be creative, you could be … you could work very hard, but not as hard as I think most of the proto-Morlocks who sit in front of their screens and just
code are forced to work. So it’s very hard. I don’t know. I mean, I’m not always sure that I can recommend this discipline to my students.

B: That will be one of our final questions, you tell me a little bit about that. I’m instructed to ask you about: (A) first, any compromises you think you’ve made in your career … things that you would have done differently, but not …

E: Well, I mean there are certainly things that I wish I had done differently. But the question is could I do it … had I known the right … this is the perfect information here, could I have known better? I’m not sure that I can identify any of those. I think … I probably, in one case, I think I probably should have left the research lab two years before I ended up going back into teaching. It was obvious to me that I was not doing what I loved and … but …

But I don’t know, I did make this decision to hitch several of my wagons to the Java star and I don’t think that that was necessarily correct. But I think … I mean, many people believed it even more strongly than I did … and it was probably the best opportunity that we had at the time. You know, it’s a question of when … for bridge players, [knowing to play for the drop] rather than take the finesse requires you to take the percentage play or peek, but, you know, that’s cheating! When someone gets lucky after doing it wrong, you don’t want to try to reinforce that behavior. People are lucky all the time in every discipline. But I think that I made strategic choices that were right.

I don’t know how this is all going to play out. You know, I’m certainly eligible to retire. I don’t know that I’m going to do that anytime soon because I … you know people are, and should be, working longer now than they did a generation ago. And I’m still … I think … you know doing a great deal of good for those students. They may be particularly the ones who are taking my courses in other kinds of things … at least serving as a role model for someone who can be interested in humanistic sides of what people think about.

The other … the course that really sort of started me on that vein at Stanford was a course that I haven’t taught for almost a decade. But a course called, “The Two Cultures: Bridging the Gap.” Drawing on C.P. Snow’s very famous 1959 essay about the split between the technological / scientific / intellectual pole on one side and literary / intellectuals on the other. I got together with a professor of English to teach a course in which we each … not just talked about the passions of our own field, but talked about our passions with the other field and how being familiar with each could generate new ideas. And since almost all real breakthroughs come at the interstitial boundary between disparate fields, being able to move back and forth in different areas and having at least some … you know, understanding of how intellectual activity has worked. I mean, having studied Latin, having, you know, been able to read philosophy, having read as much as I have, is really important to me. I would never have wanted to do anything else but that. And I think that that decision was right. And there’s partly a passion that I want to make sure that other people at least are exposed to that possibility. To know that they can be successful at an institution like Stanford and still try to be broad and not be so narrowly focused as so many people are at any institution of that sort, because we need … for a new renaissance, people who are thinking in that broadly … broad and diverse way.
B: You have kind of led into the next area that we usually like to dig into a bit. You haven’t mentioned another passion that I know you have. And that’s film.

E: Yes.

B: Is it still a passion of yours?

E: Well, I mean it is. I mean, I love film. We do still go to see, you know, many films, usually more than one a week. Lauren and I both tend to work long hours and so we get together at … for the 10:00 o’clock [pm] show of something. And we’re extraordinarily fortunate in Palo Alto in having 5 theaters that show non … you know, that are not showing just the Hollywood blockbusters. And to have a theater that doesn’t show anything other than essentially films from the 1930s and 1940s.

B: Has the Internet … did you use any of the Internet film providers, like Netflix?

E: Netflix. Not so much, because partly I actually prefer to see films in the cinema. So … and we have that opportunity, which of course most people who don’t live, you know, where David Packard used his family fortune to make this wonderful, wonderful resource, and make it available so inexpensively, you know it’s just … that’s tremendous.

The film passion I think comes out of the fact that so many of my family members worked in the motion picture industry. My great uncle is George Seaton, who directed, among other things, “Miracle on 34th Street.” And that I grew up knowing those people. My grandmother was … her role is to be the drama and dialect coach for mostly non-US actresses working in Hollywood; Ingrid Bergman, for example. And so that I got actually to know a lot of those people.

So I, you know … this was an exciting life. It was at least as exciting as my father’s faculty life. And so growing up around that and without him … my father’s brother is a director and producer, as well as my great uncle. So I liked it. But I also think that there’s something, I don’t know if it’s genetic. I made a number of documentary films, none of which will ever be released because I did the fun part. I made the film. I did not secure any permissions, which is what … you know, it’s almost like programming today, you know, that you have to spend so much of your time doing things that weren’t part of the creative enterprise. I love film editing, you know, and I’ve made re-cuts of lots of different documentary or, you know, sometimes other kinds of footage, so to make films, mostly about political history or music, history, folk music. So that those have been a passion, but I haven’t done much with it professionally.

Every course I teach outside the field certainly neatens up what’s in the field … have films that go along with them, because I tend to know the films that might apply. But that’s the only way in which it’s been part of my computer science work. I despair of the paucity and poor quality of the visionary films in computer science. I think we need more, but I don’t have the time or the connection to resources that would be possible to make just the much
better set of … and I don’t think I have the talent to you know make good visionary films about the future of computing.

B: You’ve started to touch on it, but let’s finish on the outside interests. You and Lauren have no children.

E: No, we have no children.

B: How has that affected your life?

E: Well, I mean obviously I was …

B: Has it allowed you more freedom to …

E: Well, of course it does. This is a choice that we didn’t … you know, it was in some ways made for us, you know, medically. But … I certainly thought when I was younger that I would have children. But you know, it wasn’t possible. And so we decided to make the best of that, of course.

B: I don’t see how you would have had time. Your days are so full!

E: Well, surely not having children has made it possible to pursue with the single-mindedness that I like for other things. And one of the things that we’ve been doing recently is trying to connect better with our nieces and nephews. Lauren has no surviving family at all, but I have two siblings, both of whom have three children. And that makes two apiece; no population growth there! So what we’ve done is we’ve been inviting one niece or nephew over to the U.K. for a week. We have so many frequent flyer miles that really …

B: How wonderful!

E: … we just do that. So we have a nephew coming this summer and one last summer and a niece the summer before that, since we live in England each summer. And we get to show them around London and Oxford and usually some other part of the country. And our visitor this summer really wants to visit Liverpool, not just because the Beatles were from there, but because the Liverpudlean football team is very — soccer team, of course, depending on where this is being listened to — is … has this very intensely loyal local fan base and he saw news about that. And that’s interesting because I know nothing about it.

[125:30]

B: Enriches your life.

E: And I’ll get a chance to see that. My niece Kate was in London and the kinds of things that she wanted to see were … I mean, I thought I knew … I do know London well, relative to what any American is likely to know … but you know, I had never been to Madam Tussauds waxworks — Why would one go? I had no idea! — but that’s what she wanted to do, so as a … you know? So it was fun.
B: Wonderful. Two last things, and they are both very much related. You’ve read all this science fiction and you’ve been part of future visions. In two paragraphs or less what would be your Utopian vision of the way that computer science education is going to go?

E: The Utopian vision of the way it’s going to go?

B: No. You have control, Utopia is coming, what is it?

E: For computer science education. Well, certainly we need to find a way to change. You know, I think that academia needs to put more energy and, I think, consortia of company and public investment needs to go into finding ways of making the practice of our discipline more fun than it has been recently. You know, ways of building better tools and then supporting those tools. I think the biggest problem facing academia is that we can’t support good tool development. I mean, if the fiber … I mean, the problem with the Java Task Force libraries is that it’s a full time job or several to maintain them. And if you … had there been more of a … if they’d been more successful, maybe we could have found a way to get companies to pay for that. But I really think that non-market entities have to pay for that. It’s not going to be institutions but maybe we will get federal or state support in better times; nothing’s getting paid for now. Were it possible to take the ten best pieces of software or software systems that are produced in the universities and turn them over to some company that’s compensated by grants to keep them running, it would make people much more likely to develop those things.

Open source software works in many disciplines, but I think it’s been singularly unsuccessful in computer science education because the skill base among — in terms of programming — among a lot of the faculties isn’t up to doing that, sad to say. And so I think that having those tools available and supported and … we see it in some cases. We just, of course, missed the Alice and Scratch face-off here [at the ITiCSE conference where the interview was conducted]. We need more of those things. Alice is having — I’m revealing state secrets here — trouble getting the next version out and making it stable. I think everyone knows that. And if you talk to the insiders, you know that there’s a crisis in terms of supporting that kind of thing. Now, Scratch comes out of an institution that is famous for being able to roll over, you know, the next generation of hacker into sustaining …

B: Somebody might be listening to this five years from now. Maybe you might want to talk a little bit about …

E: Well, you have at the Media Lab at MIT the … a group of people that are extremely creative and have done a lot of work going … dating back all the way to the project LOGO. First, you know, attempt to use technology in teaching, you know, at a very young age, the ideas of algorithmics. And I think that Scratch, as much as anything, is the inheritor of that base at the Media Lab and Mitch Resnick is really good at what he does. I have tremendous respect for both the people on the Alice team and the Scratch team. And they’re doing wonderful stuff. Scratch, of course, uses a more open source idea. I mean, one of the things that’s so amazing about that is that contributions from the user community, which then are available for other
people, and the Web 2.0 style of implementation. I think that has greater possibility for
momentum. But it can … and, sad to say, the Alice effort was certainly given more visibility,
if not, you know, more resources, through the tragedy of Randy Pausch’s death. You know,
everybody knows about this work a little bit more than they would have otherwise. But it
would have been better had the National Science Foundation said, “Wow, these things are
really good! We will keep them running somehow.” And let people move on to other things.

You cannot afford, as a creative researcher, to undertake things that will only be millstones
you know for the next, you know, decade if they are going to survive. And that’s a perennial
problem we … for example, in one of our iterations of the introductory course there was a
piece of software, I won’t mention which one, that was getting a lot of really good publicity.
And we decided that we would go with it. And a year later, it had vanished from the scene
because no one was there to provide that support.

One of the Utopian visions is the idea that we need a Samurai class that’s the Bacon’s Utopia
or modern … H.G. Wells’s modern Utopia for that matter, both have a house of Solomon or
a Samurai class that will keep things running. And that, of course, needs to be paid for, it’s
not clear where the revenue stream is, but it’s a public good. And there’s been no focus on
providing public goods in the educational community. We absolutely need them. And so if I
had to pick one thing that would be it.

B: Thank you. The last question you started to answer, so I might even be able to predict
your answer, but if you had advice to give a young person who is thinking about going
into computing … one short …

E: Well, the fun is really there. I mean it is possible to do more magic in computer science than
in any other field I know. I mean, it’s just … with … starting with nothing except the talents
that you bring and the creativity that you bring to make amazing things happen is
unparalleled. So, of course if you can do that, that’s absolutely what you ought to do. But the
advice is to make sure that it stays fun. If it isn’t fun, then you need to find a way to make it
fun again. And that sort of restoring the passion, beauty, joy, and awe that we’ve been talking
about ever since that first came up at SIGCSE some years back. This is, I think, the key to it.

B: And finally, if there were one story that you — about you or your career what — that
you would be remembered, everyone would say, “Oh, Eric Roberts, yeah!” What would
it be?

E: Well, I expect it to be the Curriculum 2001. I mean, I think that that worked. I was very
happy with the way that that worked. And it’s still getting a lot of use. Less in the United
States, I think, than in the rest of the world, even though we wrote it with a fairly US
perspective that was unavoidable, given who we were. So I think that that’s what it is now. I
mean, I’ve always believed that this book that I’m working on and have been working on for
almost a decade … to combine in with some narrative flow all the things that I think are the
greatest ideas in the computing world, would be my best contribution. But I have to find
ways of making it real. I mean, one of the decisions, to use Java in that, which I have
abandoned now because I don’t think it’s right, meant that so much of the work that I had
done needed to be re-done and the re-tooling.

Maybe even going back to that earlier question about what we need. I think … I’m becoming
convinced that it’s impossible to have the full range of specialties in a single individual that
one needs to really make the kind of progress that we should be making. I’m an expert, I
mean, just really a coding expert in several different languages. Certainly in C and C++ and
Java, those environments. I feel that in order to build a successful textbook today, I cannot be
merely a competent user of PowerPoint, but need to be an expert in it. And I don’t mean
some visual technology. I mean that Microsoft product, because every publisher wants there
to be slides that work in what has become, by default, and by the network externality that
favors the leader, you have to provide those with a textbook today. And if they are going to
work for a relatively, you know … an overworked community college faculty member who
is teaching three preparations per quarter in California, or five, you know, whatever it is, you
cannot afford to put your own materials together, you want good ones. So as the author I
have to be, I think that I have to be that. I believe that I have to be an expert at web
development in order to have good resources for … and you know up to a point, there’s a lot
excitement in learning those technologies. But there’s too much to hold in my head and to
keep current. So anyway… that’s gotta … I mean, I would like that to be my legacy when I
get it done.

B: Well, Eric, we spent a long time and I thoroughly enjoyed this. We really thank you for
being part of the project.

E: Well, I’m flattered to do it.

B: Thank you for the interview.

E: OK. Well, thank you, Barbara for doing it. I think this is important.