Oral History of Joseph (Joe) Traub

Interviewed by:
Prabhakar Raghavan

Recorded: June 6, 2011
Mountain View, California

CHM Reference number: X6067.2011

Prabhakar Raghavan: We are here [at the Computer History Museum on June 6, 2011] with Dr. Joseph Traub, one of the distinguished pioneers in computer science, and it's my honor and pleasure to ask him a few questions about his illustrious career. Let's dive right in. Joe, where were you born, and where did you grow up?

Joseph Traub: I was born in Karlsruhe, Germany in June of 1932. As you know, Hitler became chancellor of Germany in 1933. My father had a senior position at a bank in Karlsruhe. But in '38 the Nazis seized the bank, and it became time to emigrate. So we left Karlsruhe, went to Paris, and then to LeHavre, where we boarded the S.S. Washington of the United States Line. I still remember seeing that as we boarded the ship all the crew were lined up, all dressed up, and there were bright lights. I'd never seen anything like it. We got on the ship, touched harbor at Southampton in England, and then crossed the Atlantic to New York City. This was January of 1939, so a very stormy voyage. We arrived in New York City in February of 1939. I knew we were about number 5000 on the US quota, and I asked my father, "How did you know? You had to match when our quota number was ready, and you had to book the voyage. How did you know?" He said, "Well, I figured that was about the right time." If my father hadn't done everything perfectly, I would have been in a concentration camp. I think that helped shape my character: that if I didn't do things right, very bad things would happen. [That was my closest call until my wife, Pamela McCorduck, and I were trapped at Tiananmen Square, June 3–5, 1989.] What's perhaps somewhat surprising is the following: My wife Pamela McCorduck was born in England in 1940. She went through the Blitz. Many of my relatives were killed in the Holocaust. And yet both of us are very comfortable being in Germany. We had two wonderful sabbaticals in Munich, hosted by Wilfried Brauer, and I've organized 10 meetings at Schloss Dagstuhl. In fact, the 11th is already scheduled for September of 2012. So, spent much time in Germany.

Raghavan: Did you mostly grow up in New York, having arrived in New York in '39?

Traub: I grew up in New York City.

Raghavan: You grew up in New York City. As you were growing up, did you or people around you notice any special abilities in you as a child and a boy, growing up?

Traub: Yes, I had a knack for numbers. To give an example: I was in Germany, I was about four or five years old, and I was allowed to crawl into my parents' bed at a certain time in the morning. And I did that. Just my mother was there, and I said, "Father must have gotten up very early." She said, "How did you know that?" I said, "His alarm is set for 3:00 A.M." So my mother got very agitated, because she knew that meant that he had left early to take the train south from Karlsruhe and was crossing the border into Switzerland. He was smuggling valuable items, like Leica cameras-- you know, something small and valuable like a camera, platinum cigarette case -- already hoping that then it could be turned into money, so when he felt it was time to emigrate he'd be able to do that. I taught myself how to tell time: hours, minutes, percentages, while we were still in Germany. In the eighth term of elementary school, we started being taught time and percentages. I was bored out of my mind. My father, as I said, had a senior position at a bank. My mother was a housewife, but she was also very facile with numbers. In the United States, we would sometimes compete in mental arithmetic. She was at least as fast as I was, so I seem to have inherited this from both parents.

Raghavan: Let's move to the US. You were in New York, then you went to high school. What was that like?

Traub: That was eye-opening. I went to a mediocre elementary school. As I mentioned, we were still learning how to tell time in eighth grade. But I was able to get into the Bronx High School of Science, by examination, and that was an eye-opener. That was a wonderful place, and that's when I first found out how many smart kids there are in the world.

Raghavan: You talked a bit about school, and the Bronx science school. What about fun? What did you do for fun back then?

Traub: When I was about 12, my uncle taught me the rules of chess, and we would play chess exactly once a week: every Saturday. For six months, I never won a game. After six months, I never lost a game with my uncle. When I was 14, there was a park near where we lived, and there were a bunch of rather strong chess players. I played against them, so I got much stronger that summer. In particular, there was a character named Irving. If you took too long thinking about your next move, Irving would say, "Move or pay rent", so you moved. [I'd set up a board at home and play against myself. After making a move I switched chairs and played the other side. It made me aware of the "other" person's strategy, which carries over to other things in life.]

The following fall, when I was in my second semester at Bronx Science, I made the chess team. A couple of years later, I was captain, and first board of the chess team. And very active in chess, until I was about 17, and then I just dropped it. I did that several times in my life: I would be very involved in something, and then just burn out on it, or get tired of it, and drop it. Also, girls didn't play chess. What are some of the other things? Reading. I loved reading, so much so that my parents rationed me to one book from the library a week. So I asked my parents for a flashlight for my birthday, thinking, "Oh, I'm going to read in bed, under the covers, with a flashlight." Well, that doesn't work very well. So, reading. Travel. These are some of the things I do for fun, and continue to do. Travel: after I got my Ph.D., my first wife, Susanne Gottesman, and I spent the summer, 14 weeks, touring Europe. I particularly loved Spain. I'd never been in the American West. I didn't realize how much Spain was like the American West, or the American West is like Spain in climate and in topography. I still love to travel. That trip to Europe got me interested in art, got me interested in food, both of which I'm still interested in. Early on, probably beginning with the Second World War, I became a news junkie, and followed the news avidly, and still do. Enjoy hiking. In fact, one of the pleasures in Santa Fe is to go hiking in the mountains behind Santa Fe. Some of the things I do for fun.

Raghavan: So, by the time you got to high school--

Traub: Oh, I should mention two other things-- very significant, in terms of fun-- that's tennis and skiing. In my teens I discovered tennis; a bunch of us played tennis on city asphalt courts by the Hudson River. I didn't even reach the level of mediocrity in tennis. Later on, when I took lessons, I still never got to the level of mediocrity. Skiing was a little better. My father used to tell me stories of skiing in the Black Forest, because Karlsruhe was at the edge of the Black Forest. He also went skiing in the Alps in Austria and Switzerland. Of course, in those days, there were no lifts, so it was all hiking, climbing uphill, and then skiing down. My first wife Susanne took me skiing, a place called Bromley, in southern Vermont. I just loved it. I loved everything about skiing: in blizzards, I loved the equipment, I loved the ski lodges. I skied, I guess, for something like 50 years. I finally stopped in the 70s, when the snowboarders took over. They're rather reckless, and a bunch of my friends got very serious injuries. I said, "Oh, I think I'll hang up my skis." I never got past the stage of being a strong intermediate, but I was a decent skier.

Raghavan: As you got through high school, computers were coming into being. When did you first come across computers, and what was your first real exposure to computers?

Traub: Pure, dumb luck. I took a degree in math and physics, at CCNY: City College of New York. And then entered Columbia in '54, intending to be a theoretical physicist. In 1955, my best friend, Sheldon Penman, said, "You should go over to 612 West 116th Street. There's something there that will interest you." What was there was IBM's Watson Labs. I got hooked on computing. Many years later, in fact fairly recently, I was in Cambridge, and had dinner with Shelly, who went on to become a professor at MIT. I said, "Why did you send me… how did you know that computers were the thing for me?" He and I used to study together. He said "Because you could solve any problem." Now, I don't remember this, but that was what he claimed. I think our partnership was: he would develop the model, which I had never been good at, and I would solve it. So that's why he sent me to Watson Labs, and that's where I discovered computers.

Raghavan: You were in the Physics Department then. Did you, as a result of the exposure to IBM Watson, leave the Physics Department and enroll in a different department, or what did you do?

Traub: Well, I left physics, but I didn't enroll in a department, because there were no departments of computer science, and I wanted to study computing. There was a committee on applied mathematics at Columbia, and it consisted of people from mathematics, statistics, physics, engineering, perhaps some other departments. I got my degree under that committee. It's interesting; I was never in a department -- except as an undergraduate, which doesn't count-- I was never in a department 'til I became head of the Computer Science Department at Carnegie Mellon. I think that was actually an advantage, because I was willing to try anything; I had no preconceptions of what a department should do.

Raghavan: As you went through your Ph.D., how did you choose a topic, when there was no computer science department? What was the topic, and how did you choose it?

Traub: You remember, I left chess when I was 17; that was 1949. But I ran across a man named Alex Bernstein, who, in fact, was a pioneer in computer chess. He was working at IBM, and he had been on the chess team at Bronx Science a couple of years before me. Alex invited me to work with him, and do my thesis on computer chess. I went to my advisor, who was a professor of mathematics, and said, "Will Columbia give me a degree for computer chess?" He gave me a one-word answer: "Never." That was probably a life-changer, because if I had done that work on computer chess, I would have been very early into AI [Artificial Intelligence], and I might have gone down that path. I was told “no”: not computer chess at Columbia.

There was a professor of physics named Henry Foley. Foley suggested I compute certain quantum-mechanical numbers in excited states of helium. The experimentalists knew those numbers much more accurately than the theoreticians could compute them. So my thesis was computational quantum mechanics. What helped make that a Ph.D. thesis, I should say, was that I was doing it on an IBM 650, which, as you know, is a 2000-word, drum-memory machine. [The need to be economical on this machine may have led to my early interest in optimal algorithms and computational complexity.] That was part of the challenge.

I had a Watson Labs fellowship, and that gave me unlimited access to the IBM 650. It also gave me a stipend, and an office. But particularly, it gave me access to the machine. I was running 10 hours a night. The 650 was like a personal computer, and I would run it all night, get 10 hours of computing. But that wasn't enough; this was a big calculation. I managed to get access to two more 650s, and I would run three of them, simultaneously, through the night. I'd decompose the problem, and get 30 hours of computing through the night. One of the 650s was at a secure facility at 125th Street. I would run in with my box of cards, and do something with the machine -- maybe load some new cards, take off cards, and run out again. This was a secure facility; you had to sign in and sign out. That must have upset somebody, so I was denied access, and I was down to only two machines. Discovered that I loved programming. That was the beginning. I should say that the entire thesis was published in *Physical Review.* Two papers, which I looked at recently. A lot of work! [It was while working on my thesis that I realized I had a gift for research. I was determined to overcome every obstacle.]

Raghavan: Having done this thesis on computational quantum physics, how did you then decide to make computing, and eventually, computer science, your career?

Traub: I remember the moment very distinctly. I should say, that as part of doing my thesis and running between three machines, I had a very high-tech way of transporting data and programs [between the machines]: I had bought a used car. I think I used the money from my New York State Regents scholarship. I had a used car, I had boxes of cards, and I would drive from facility to facility with my boxes of cards. That was my high-tech way of doing it. I'd been programming and testing for six months to get the beginning of these numbers that I wanted. This was almost like a test run. I got the cards off the 650, and I carried them over to the printer. I was doing a variational calculation, so the energy was dropping. That's what I expected to see. I loaded the cards into the printer, and the first numbers that came off the printer were the energy of the particular state I was looking at, which agreed with the measured energy to something like four decimal places. There was a chill going down my spine. I couldn't believe it. That was the magic moment.

Raghavan: Then there was computing as a career, after that.

Traub: Yes. But that was the beginning. That was, I think, the key moment.

[This might be a good moment to mention the pivotal role that IBM has played in my professional life. As I mentioned, I had a Watson Labs Fellowship, which gave me unlimited use of the IBM 650. In the late 70s, I started the CMU-IBM connection. When I started the Columbia department, a $400K gift from IBM was invaluable. Five years later, IBM gave the Columbia CS Department a much larger gift. The key person behind the gifts to Columbia was Senior Vice President Ralph Gomory.]

Raghavan: If you look back at all the research you've done, which spans a wide spectrum, tell us a bit about some of the early research that you did at that time?

Traub: [One thing I did not want to do is continue working on computational quantum mechanics.] Just like with chess – after I had done chess very thoroughly, being quite obsessed with chess for a while – after spending three years on the thesis I wanted to get away from that. Wasn't the least interested in continuing that work. I actually had an invitation to get a postdoc from somebody who was working in this area; was absolutely not interested. [I should add that with my current interest in quantum computing, I've returned to solving problems in computational quantum mechanics.]

I'd been hired by Bell Labs, and showed up there in September of '59. Very soon, somebody came to my office and had a complicated nonlinear equation to solve. He asked me, "What's the best way of solving this?" I thought about it, and I could see a bunch of ways of solving it. I thought, "But there must be a best method, an optimal algorithm, for solving this problem. There must be a theory of optimal algorithms." So I did a bit of searching the literature. Well, there was no theory of optimal algorithms in 1959. Computational complexity wasn't known; it was started by Hartmanis and Stearns in '65. Kolmogorov had given a talk [on the topic] – actually I was at that meeting -- at the International math congress in '62, but it was in Russian, Kolmogorov was not a good speaker, and I didn't understand the impact. So, essentially, there was no computational complexity in '59. I advised him how to solve his problem, but I set about creating a theory of how do you solve nonlinear equations. The key insight was that if you're solving f=0, all that matters was the information you had about f. For instance, if you knew f and f’, and you confined yourself to methods that evaluated the function and derivative at one point, you could not beat order of convergence 2, by any algorithm, [independent of the structure of the algorithm]. And this is true in general. I created other families of methods, multipoint methods, one point with memory, etc. [I proved what I called the fundamental theorem of one-point iterations.] But the key always was: information. It turns out the information determines the maximal order, and maximal order is related to the computational complexity of that problem.

So I had a mass of material. I sent something like a 120-page paper to Mario Juncosa, the editor-in-chief of the Journal of Complexity. I was very naïve. This was around 1961. Mario wrote me back and said, “It’ll be a long time before I read this paper.”

Just then, Prentice Hall approached me about writing a book. A book, to me, meant immortality. A book! So I devoted myself for some years to creating the theory and writing a book about it. It was called "Iterative Methods for the Solution of Equations." George Forsythe was the editor-in-chief of the series in which this book appeared; I think it was the fourth book in the series. As you know, George later, in 1965, became the founding chair of the Computer Science Department at Stanford. The book was published in '64. It's still in print almost 50 years later. I think the American Math Society now continues to issue it, and it's sort of become the standard book in the field. The name of the book was wrong; it should have been called "Optimal Iteration Theory." That was the first major piece of work I did. [It was the beginning of a lifetime of work on optimal algorithms and computational complexity. The introduction to the book begins, “The general area into which this book falls may be labeled algorithmics.” A 1979 paper by Don Knuth credits me with coining the word.]

As you know, for solving a general nonlinear equation, you can only guarantee convergence under certain conditions. So I thought: is there a class of nonlinear equations for which I can guarantee global convergence? It turns out, the answer's yes. It's for polynomial equations. In '66, I published the first paper on global convergence [for polynomial equations]. Also in '66, I was on leave from Bell Labs, and I was at Stanford, where I met a Ph.D. student named Michael Jenkins. Mike and I started working together. Michael and I made major improvements. We introduced the idea of a shift, which was crucial, and we published that paper in 1970. Michael wrote a high-quality, portable piece of software, and that has been one of the standard ways of solving polynomial equations, both because the program was good and I think the algorithm was rather good. It’s generally known as the "Jenkins-Traub algorithm," or, as I say modestly, when I teach about it, the "Jenkins-T algorithm.

Raghavan: It sounds like you had a fairly fruitful career at Bell Labs early in your life. What caused you to leave Bell Labs?

Traub: I was at Bell Labs in the '60s. It was a golden time. If you were hired into the Research Division, Division 1, you could do anything you wanted. There were ample funds for computing, for travel. Very good colleagues. Really, a golden time at Bell Labs. If you had impact on the company, that was all the better. A wonderful time. And yet, somehow I knew that I would eventually wind up at a university. In 1969 Pamela McCorduck and I got married. Pamela and I visited a number of universities. I accepted a position as full professor with tenure at the University of Washington. I met with Ed David, who was my executive director, and told Ed I was leaving. Technically, I think, I was on leave from Bell Labs, but that I was leaving. He said, "Oh, leaving Bell Labs; that's a very serious matter, very serious decision." Well, shortly after, Ed David also left Bell Labs. He went to Washington as the President's science advisor.

Raghavan: How did you wind up at CMU, and what was your early research there like?

Traub: I was at University of Washington in 1970, and in the spring of '71, I was invited to interview, by Al Perlis, for the position of head of the department at CMU, and I was asked to serve there. I was 38. That's part of being in the field so early; in a mature field, I would not have had such an opportunity at age 38. The first work I did at CMU was the following: I had been invited to be a consultant at Lawrence Livermore Labs, because Livermore was expecting a [Control Data Corporation] STAR computer, which was a parallel vector machine. That got me interested in parallel computing, but there was very little interest in parallel computing, because there were essentially no parallel computers at the time; the STAR was one of the first. I remember giving a talk at a very famous computer science department at a very famous university, and I talked about parallel computing. The senior professor who was in charge of the seminar where I spoke asked the first question. He said, "Joe, you don't think we're ever going to <u>use</u> these machines to do our computing?" Well, I thought, in fact, we would. In 1973, I organized what may have been the first symposium on parallel computing, at CMU.

In the meantime, something else happened that was to be a life-changer. There was a very talented, very powerful young mathematician in Warsaw, named Henryk Wozniakowski. As you know, at the time, there was a Communist government in Warsaw. Communication with the west was very restricted. But Henryk was able to get to Germany for a visit, and he carried with him two papers. He sent them to me registered airmail, because he wanted to be sure I would get them. He told me much later: sending those two papers airmail and registered cost him a month's stipend! But he wanted to be sure I got them. I did, and that was to be a life-changer for both of us, because they were two wonderful papers. I had made some conjectures in my '64 monograph. I should tell you that Henryk got interested in optimal iteration theory by reading that book. He not only proved the conjectures, but he did it in a very general way. I was just doing it for univariate functions [scalar equations], and he did it for abstract spaces, a special case would be systems of nonlinear equations. Very general. A very powerful advance. I invited him to come to CMU. He arrived in 1973, and that was to be the beginning of a 40-year collaboration. A very rich collaboration. Henryk and I continued to work on optimal iteration theory until 1976, when there was to be a total change in our work. We stopped working on optimal iteration theory.

Raghavan: What happened next, in 1976, and what change transpired?

Traub: We had a Ph.D. student named Arthur Werschulz. He's now a professor at Fordham University, and also involved with our research group at Columbia. Werschulz gave a seminar where he used the techniques we had developed for solving nonlinear equations, and used it to attack the problem of integration. Well, that's a very different problem. A nonlinear equation is solved by iteration. An integral is of course a linear functional; [to approximate an integral] you don't do iteration. My reaction was: there had to be a deeper structure to tackle both these problems. Henryk and I always had a list of open problems, and we called this problem -- what's the deeper structure and the questions about the structure -- the "S" problem. The "S" stood, perhaps, for "special," perhaps for "structure." We started working on this, and Henryk Wozniakowski and I published [our results] in our 1980 monograph. That was really the beginning of information-based complexity. We developed the theory over abstract linear spaces, like Banach and Hilbert spaces, but the applications were for high-dimensional problems.

We called this new field "analytic complexity." The reason for calling it "analytic complexity" was to differentiate it from "algebraic complexity", which, in fact, I'd been working on. Let me backtrack for a moment, because I want to talk about algebraic complexity. The beginning of algebraic complexity was due primarily to the work of Winograd and Strassen in the late '60s, on matrix multiplication. I got interested in algebraic complexity in the early 70s, and worked on three problems. The first was the following: how fast can you compute all the derivatives of a polynomial of degree n? Now, it's known how to do this. It goes back to Horner some 200 years earlier. Horner was interested in this because if you translated the polynomial, which was used for his method of solving polynomial equations, you had to compute all the derivatives. I covered this in my course [at CMU]. I said it's a very elegant algorithm, very simple, it takes order n squared multiplications [and additions]. And, I guess it is optimal – I just tossed that off. A woman named Mary Shaw was in my class, and Mary said, "I can do better." We started working, and we drove the number of multiplications and divisions, because we now needed divisions, down to linear [order n]. That was called the Shaw-Traub algorithm. Mary was actually in software, but I think she was very proud of this theoretical result. She is now the Alan J. Perlis Professor of Computer Science at Carnegie Mellon. [Incidentally, there are still several interesting, and I think hard, open questions in this area.]

The second algorithm was work done with my Ph.D. student H.T. Kung, who is now a chaired professor at Harvard. Kung and I had the following result: If you want to compute the first n terms of an algebraic function, you could do that at cost the same as multiplying two nth-degree polynomials. This problem had a very illustrious history. Isaac Newton worked on it. But Newton missed a key point, so he didn't get our result. He didn't realize that it was guaranteed quadratically convergent, symbolically. So we had this very nice result: you can compute the first n terms of any algebraic function and it costs you no more than multiplying two nth-degree polynomials.

The third of these results was work with Richard Brent, who is a professor at Australian National University. I think he's one of those [few] people, together with Bob Tarjan, who'd gotten through Stanford [with a PhD] in two years. Richard and I worked on the following problem: You want to compute [the first n terms] of the qth composite of a power series [for any q]. Our result was you could compute the qth composite at cost no greater than computing a single composite, for any q. For instance, if q is 1/2, you're computing the square root, etc. Amusing anecdote: Donald Knuth called me – these were the days before email, so he had to call me – and Don said "I've heard about this result; what is it; this is my understanding of what you've proven." [He] wanted to talk about how we proved it. I said, "I can send you a preprint in a few days. We're very close." Don said, "No, I'm working on that part of my book right now." He wanted to tell me how he had solved a certain part of the analysis, and whether it matched what we had done. I said, "Yup, that's the way we did it." It appears in Volume 2 of "The Art of Computer Programming." But I think it shows something about Don. Hearing about the result, he did the analysis to get that result, even though he could have our analysis in a few days. If it were me, I would never spend my time deriving something that somebody had already done; I just wanted to do something new. But that's why Don is so good. He said, "Ok, I want to understand this. I'm going to do it myself."

Forward to Werschulz's lecture, the looking for the deep structure for continuous problems, [Wozniakowski and I] published this monograph in 1980. We called the field "analytic complexity" to differentiate it from, "algebraic complexity." But I wasn't entirely happy with that name. In '83, Greg Wasilkowski, who'd been a student of Woźniakowski, joined us and we published another monograph, and we now called the field "epsilon complexity." "Epsilon" was because you couldn't solve these problems exactly. They had to be solved numerically. You could only solve them to within some small parameter, hopefully small, called epsilon. Pamela asked me, "Why's it called 'epsilon complexity'?" and I said, "Well, epsilon's a small quantity and indicates the error. We want to emphasize we can't solve the problems exactly." Well, Pamela was not enthusiastic about this explanation. Since Pamela has written a bunch of books and I have lot of regard for her command of the language, I was looking for a better name for the field. I mentioned this to my friend Richard Karp, one of the pioneers, as you know, of NP-completeness, and explained the nature of the new field. Richard suggested "information-based complexity." That's what we adopted as the name of the field. That was the name of our 1988 monograph. We usually call the field "IBC." It deals with optimal algorithms and complexity for continuous problems, which are typical of much of science and engineering, and even finance [and biology].

Raghavan: When you came to this point of defining information-based complexity, what did you start to see as the applications of information-based complexity?

Traub: The applications are very important to me. Applications are how you have impact, often. A prime example is Claude Shannon's work on information theory. He solved a problem people in communications really needed to have solved. Well, our applications were not in that class, but I can mention a couple. In the early '90s I was able to get a collateralized mortgage obligation, a CMO, from Goldman Sachs. I had a student, [Spassimir Paskov], who was very strong in theory, but I wanted him to get some experience in hardware and software. So I said, why don't you take the CMO, and you have to compute something like ten integrals [for the ten tranches] in 360 dimensions – the number of months in 30 years. And why don't you do it on a workstation farm – whole bunch of different workstations. So it involved some hardware; it involved some software. Why don't you run [and compare] quasi-Monte Carlo and Monte Carlo on these integrals. Now, quasi-Monte Carlo, as you know, uses deterministic sampling;

Monte Carlo uses randomized sampling, It was well known by the experts in the field, and they were very good, that you should not use quasi-Monte Carlo for integrals with dimension greater than 12.

Raghavan: Makes sense.

Traub: That was well known. The reason was that there was a certain theoretical bound, and it said really bad things would happen for high-dimensions. So Paskov ran this, and he came back to say quasi-Monte Carlo always beats Monte Carlo by one to three orders of magnitude. The reaction of my research group was, “No, that’s impossible. You’re making a mistake. In fact, we expect Monte Carlo to be at least as good, maybe better.” I held up his Ph.D. six or nine months, because I wasn’t convinced. Finally Paskov convinced me he was right. Paskov and I went to Wall Street, and talked to the usual suspects -- Morgan Stanley, Citibank, etc., -- and they were also not entirely convinced. Then other people came along and corroborated our results. I think people on Wall Street often use quasi-Monte Carlo, as well as Monte Carlo, to do computation.. [After Spassimir graduated, my wonderful colleague, Anargyros Papageorgiou, spent part of his time on computational finance.] I should say that quasi-Monte Carlo is not a panacea for all integration, and a small cottage industry has grown up around the question, “What is it about computing financial derivatives? What characterizes the problems for which quasi-Monte Carlo beats Monte Carlo? In particular, why is it better for financial derivatives?” That’s still open. So there was one application, to computational finance.

The second was to quantum computing. In 2001, Erich Novak, who is now at Friedrich Schiller University in Jena, published the first convincing paper, with algorithm and complexity analysis, on computing integrals on a quantum computer. That opened my eyes. He used IBC techniques. To me that said, “This is an interesting, new rock to turn over.” I always try to get into fields early: parallel computing early; algebraic complexity pretty early; [co-founding] information-based complexity; now quantum computing, but for continuous problems. There had been Peter Shor’s result for factoring numbers [a problem with complete information], but this is for continuous problems. We started working on that. We [Henryk and I] looked at path integration [on a quantum computer. As you know, path integration (due to Dick Feynman) is an alternative formulation of quantum mechanics]. [Anargyros Papageorgiou and I, together with two of our graduate students] looked at solving the time-independent Schroedinger equation on quantum computers using IBC techniques. [We showed that a p-particle system could be solved with cost linear in p using p qubits on a quantum computer]. So there are two applications of IBC. I’m still interested in the financial problem, because we still don’t know why, really, quasi-Monte Carlo beats Monte Carlo for financial derivatives. And, of course, I’m continuing work on quantum computing.

Raghavan: Joe, we’ve talked a fair bit about your research, but you’ve also been known to build organizations, so could you share with us a little bit about the organizations you built?

Traub: As I mentioned earlier, I became head of the department at Carnegie Mellon in the spring of 1971, showed up July 1 of 1971. Alan Perlis, who was the founding chair, was leaving to become founding chair at Yale. He and I only overlapped for a few days, but he gave me some advice which was invaluable. One of the pieces of advice was: “Tenure Bill Wulf as soon as you can.” We did that. Soon he became full professor. Later he became the second chair, succeeding me, at CSTB [the Computer Science and Telecomunications Board], and still later went on to become president of the National Academy of Engineering. Now, about half the faculty had left CMU the previous few years, gone to other departments. But the people who were left were a very strong group of scientists and engineers: Gordon Bell, Nico Habermann, Alan Newell, Herb Simon, Raj Reddy, Bill Wulf. But it was important to build [the faculty]. Working with the faculty I was able to get some very good new people. My budget was very simple. I had a $2 million budget. $1.8 million was from DARPA and $200,000 from Carnegie Mellon. That left us rather vulnerable, so we worked to diversify the funding. It was a wonderful department. A wonderful group of people to work with. [Newell, Simon and I began talking about the greening of CMU and Pittsburgh through computer technology, and that has come to pass big time.] By 1979 we had some 50 teaching and research faculty. I’d been head for seven years, and I could think of new challenges.

At that point I was approached by Peter Likins, who was the dean of engineering at Columbia. There were two groups [trying to build computer science] at Columbia, one in electrical engineering, and the other in statistics. They were at loggerheads. They would try to recruit, they visited universities, they had people in, but they were undercutting each other, fighting each other, and nobody wanted to come into an environment like that. So, the decision was made by Columbia to terminate both those efforts and start a new department. Peter, who’s a wonderful man, recruited me to be the first head.

Peter Likins is a remarkable man. He became provost at Columbia -- very unusual for a dean of engineering. Then he became president of Lehigh, and then president of the University of Arizona, recently retired. Peter persuaded me to come to Columbia. Of the institutions I have built, that was the most difficult. To give you one example, at Carnegie Mellon we had enough computers to heat a very large building. At Columbia the entire engineering school had a single mini-computer. As I remember, it was a DEC 11/45. No connection to the ARPANET. I inherited three tenured faculty from the discontinued efforts, and a bunch of young faculty who did not belong in a department with national ambitions. They, the non-tenured faculty, were all gone within two years. Now one of the jobs was to hire people. We had 2,000 students taking our courses at this time [and almost no faculty.] You couldn’t get your hands on a computer, generally – remember this is 1979 -- except through computer science. I was hiring young hotshots, the best young Ph.D.s I could get. I should say, the competition was very rough in ’79, because we were competing against MIT, Stanford, CMU, Berkeley and Cornell, etc., and in the New York area against IBM and Bell Labs, both of which were going strong. But we hired some very strong young people, and I did not advertise to them that every one of their classes would have 200 students. So I hired young faculty. I took them down [to Arlington, Virginia] to meet Bob Kahn, who was the head of IPTO [Information Processing Techniques Office of DARPA]. Bob was very impressed and gave us a major DARPA contract. Also through that, for the first time, Columbia got an ARPANET connection. We got a new building. [One of my conditions for coming to Columbia was that computer science would get a new building, and I would select the architect from a list provided by Columbia’s Dean of Architecture. In 1982, we celebrated our new building with a symposium and an honorary degree for Herb Simon.] I set up a computer research facility. We started bachelor's, master's, and Ph.D. programs for all of Columbia. That was the beginning of the building of the [computer science] department at Columbia.

The next thing I started was The Journal of Complexity, in 1985. I was not interested in editing a journal, but we had a lot of papers now on information-based complexity and no natural place to publish them. To the best of my knowledge, and to my surprise, there was no journal with the name "complexity" in the title in '85. [Stephen] Wolfram started a journal [The Journal of Complex Systems in 1987.] So I started the Journal. It's now in its 27th year, and I'm still editor-in-chief. But I made the key decision to create a [strong editorial] board. This was after a while, because I realized I was the bottleneck, because all papers were coming through me. Create a strong board, where every member of the board could handle papers. The final decision [to accept or reject] came to me, but I generally went along with what the associate editors recommended. The papers for which I didn't have an associate editor or senior editor, I would handle. But that made the job quite doable. It's in its 27th year now, and is much broader than information-based complexity.

The fourth and last of these institutions was one you're very familiar with, the Computer Science and Telecommunications Board (CSTB). I was approached in 1986 about starting a computer science board for [what is today called] the National Academies. Or more properly, as you know, the National Research Council (NRC), which was the part of the National Academies that actually did the work. [I had been approached by Frank Press. Frank is an extraordinary man. He had served as President Carter's Science Advisor, and then in 1986 was president of the National Academy of Sciences, and Chairman of the National Research Council.] I knew there had been two earlier attempts to start computer science boards, both of which had failed. This was a third try. I was determined this one would succeed. I appointed a very strong board, half from academia, half from the industrial sector or corporate sector, and that has generally been the case – half and half. I was very concerned about raising money, because each board was self-sufficient, and especially core funding, because core funding enabled us to start new initiatives, pay for board meetings, etc.

About a year in, I hired a staff director named Marjorie Blumenthal, who I think became quite legendary. She was very strong. For many years she served in that position. That was a key hire. We started doing reports, and apparently we did something very unusual for the NRC: we did policy as well as technology. Generally the technology boards were expected to do technology. Well, we also did policy, including our first report, which had to do with the National Information Infrastructure and was mainly worked on by Michael Dertouzos, but it was a board report. Frank Press said to me, after a while, [that] we were probably the strongest board at the NRC. Now, in about 1988, Frank Press said BOTCAP, the Board on Telecommunications and Computer Applications, was failing. The decision had been made to terminate it, and we were to take over telecommunications. Well, we had this nice name, CSTB. It was the Computer Science and Technology Board, and Marjorie and I wanted to preserve that name, preserve our stationery. Well, we did the obvious and called it Computer Science and Telecommunications Board, still CSTB, as you know, until this date.

So, I built those four major institutions: computer science at Carnegie Mellon, computer science at Columbia, the Journal of Complexity, and CSTB. What they had in common, and the key, as you know, was excellent people.

[I'd like to mention two places that have been very important to me. One is Schloss Dagstuhl, located in the Germany countryside. The Dagstuhl Seminars were started in 1990, and we organized one of the first on "Algorithms and Complexity of Continuous Problems" in September 1991. It's an invaluable opportunity to spend a week with leading researchers on this topic from around the world, while eating good food and drinking good wine. Our eleventh seminar will be in September 2012. I've been on the organizing committee for all of them. Another magical place for me has been the Santa Fe Institute (SFI) in New Mexico. Henryk and I taught in the summer school in 1990. It's the mother church of the "sciences of complexity." What I particularly like is the high degree of interaction between people from different disciplines. I've had formal or informal associations with SFI since 1990. Currently I'm an external professor.]

Raghavan: If we were to step back and ask you what accomplishments are you most proud about, what would you say?

Traub: Well, it's part theory [research], part institution-building. I've just talked about the institutions. In theory it was the creation of optimal iteration theory [which was a precursor of information-based complexity]. Then [the creation with students and colleagues] of those algorithms for algebraic problems, of which certainly the best known is the Jenkins-Traub for polynomial equations. [But there are also the Shaw-Traub, Kung-Traub, and Brent-Traub algorithms.] Then the co-creation, initially with Henryk Wozniakowski and then with other colleagues, of the field of information-based complexity. [And more recently, the work on financial computation and quantum computing, primarily with Anargyros Papgeorgiou.]

[I'm also very proud of my Ph.D. students. I'll tell you about just two of them, H. T. Kung, and David Lee. In 1970 I was at the University of Washington. I had an ONR grant which included money for one GRA. About ten students applied and I chose Kung. I can't tell you what he gave off at the interview which made me choose him. I brought Kung with me to CMU. We used to argue about research in front of my whiteboard, and I'd say that since he was Chinese I expected some filial piety. After he got his Ph.D. I wanted to appoint him to the faculty. We almost never kept our own faculty while I was head. The only exceptions were Kung, Mary Shaw, and Anita Jones. The Immigration Service wanted to deport him, claiming there were qualified American citizens for this position. They gave as an example a woman with a master's degree who had never done any research. Incidentally, she happened to be the wife of one of our faculty. All appeals were refused. At the last moment, we got access to then-Representative John Heinz through the wife of one of our faculty who worked for him. Heinz interceded and Kung remained in the U.S. He became full professor at CMU and is currently a chaired professor at Harvard.

The other student that I'd like to tell you about is David Lee. During the Cultural Revolution in China, he taught himself advanced mathematics and other subjects. When he came to the United States, he earned his Ph.D. in 1985. I remember Pamela and I walking with him in the Embarcadero area of San Francisco. He was pessimistic about his future prospects because he was older than the other students. Fourteen years later he was vice-president of Bell Labs, and Founding Director of Bell Labs Research, China. Then he returned to the United States as the Ohio Board of Regents Distinguished Professor at Ohio State University. Recently he became Founding Director of the Networking and Communications Lab of HP Laboratories.]

Raghavan: You served as department chair in a couple of places. Were you not tempted to go higher up in academia to become a dean and provost and so on?

Traub: Well, in the mid-'70s Carnegie Mellon was doing a search for the dean of science. Today most computer science departments are in schools of engineering or are separate schools. But at Carnegie Mellon in those days it was in the School of Science. Dick Cyert, the very able president of CMU, made it clear that he would like me to be dean of science. But I felt I was so involved with computer science, I really couldn't do the job. It wouldn't be fair to the other departments in the school – physics, math, etc.. I was too much involved with my own department. So I declined. I was in my early 40s, so if I'd accepted that position, and had been successful, I might well have had a different career path. Over the years I've been approached by various institutions for senior academic positions, which I've always declined. I've stayed in the field, building institutions part of the time, but always staying in computer science. I've not regretted that decision.

Raghavan: If there were one thing you could do over again, potentially differently, what would that be?

Traub: It was something I should have done and didn't do. In the mid-'80s I started getting concerned about the information infrastructure, and how vulnerable we were. We were the most advanced country in the world in the use of information technology, and I imagined myself to be an enemy country or a terrorist -- I don't think I would have used the word "terrorist" in the mid-'80s -- but a terrorist or an enemy country. What would be our vulnerabilities? The vulnerabilities would be the information infrastructure, and also perhaps the physical infrastructure. Example: In the mid-'80s I was able to be on the floor of the New York Stock Exchange, and the only security I could see was one security guard, maybe a policeman, with a revolver. As far as I could see, that was the only security. Now, there may have been hidden security. I doubt it. My reaction was that this was the heart of capitalism. What if somebody tossed a couple of grenades on the floor of the New York Stock Exchange? It would only be symbolic, but think of the symbolic damage. The processing, as you know, was not done there. It was done across the river in Brooklyn, but I visited that facility and it didn't look that secure, either. So I was concerned about physical and electronic security. I should've gone public with this. As chair of CSTB I had access to Al Gore, who I think he would have understood the issues. In fact Al gave a wonderful talk one evening to the CSTB around '86. But I didn't. The reason [I didn't go public] was: what if I talked about these concerns, and someone actually launched an attack? How would I feel? I think that was a mistake, because our enemies are very smart. Anything I could come up with they had already thought of. So I kept quiet for 10 years. Finally in 1996, on the 10th anniversary of CSTB, I gave the keynote address at our 10th anniversary symposium, and I talked about this vulnerability. I talked about attacks on what I called “the virtual estate.” By “virtual estate” I meant bank accounts, equities [mutual funds], CDs, you name it. All of our wealth is just a bunch of electrons! What if somebody was able to attack those? Furthermore, as you know, there are all sorts of other vulnerabilities: [the electric grid,] the air-traffic control systems, the train system, water. Everything is controlled by computers. I expressed my concern, said we should be very concerned about cyber-security, and that the responsibility should be in the executive branch of the government for worrying about that. But in the meantime ten years had been wasted, and I’ve always regretted not really trying to do something about our security vulnerabilities.

Raghavan: We come to our last question, Joe. Can you…

Traub: Already? I’m just warming up.

Raghavan: …take a few minutes to summarize your life and career?

Traub: In two words: very lucky. I was lucky to come to the United States. I’m going to start crying. Very lucky to come to the United States when I was six years old. This country’s been very good to me. Lucky that I got involved with computing very early on. Lucky to have had wonderfully talented and generous colleagues. I have a wonderful marriage, loving children and grandchildren. LUCKY.

Raghavan: Excellent. Thank you, Joe.

Traub: Thank you.

END OF INTERVIEW

