Prof. Peter Wegner

Oral History by Nathan Ensmenger

May 2009

ENSMENGER: My name is Nathan Ensmenger. I'm here at Brown University with Professor Peter Wegner in an oral history undertaken for the Association of Computing Machinery History Committee.

Professor Wegner, I thought we could perhaps start by discussing your early life and education, things that you think are significant to your subsequent career in computing.

WEGNER: One of the defining moments of my life was when I survived the Holocaust. My parents were from Austria, and when I was six years old, in 1939 (I was born in 1932), I was put on a train that took me from Vienna to London. It was a special, sealed train, organized by the Red Cross, and was called a *Kindertransport*—a children's transport. This was in April of 1939, and I arrived in England about four months before the war began. This was shortly after the horrific events of *Kristallnacht* ("The Night of Broken Glass"), when the Germans destroyed many Jewish synagogues and schools. The British government—or rather, British Quakers who lobbied the government—made it possible for children to seek refuge in Britain. Children, but not their parents. As it happened, my mother was already in England, and her brother in the United States, so I had at least some family members who survived Hitler and the Nazis. But most of the other children on the *Kindertransport* trains were orphans -- or soon to become orphans.

ENSMENGER: Whom did you live with? Or where did you live when you arrived in Britain?

WEGNER: From the age of six-and-a-half, I lived in various places in England. I went to a progressive coeducational boarding school called Bunce Court School. It was an excellent school—I actually am still in touch with many of my fellow-pupils from Bunce Court.

The director of the school, Anna Essinger, a Jewish woman who had become a Quaker, transferred the school from Germany to England in 1933 when Hitler came to power. She recognized that the school, although not strictly Jewish, had many Jewish pupils, and therefore could not safely remain in Germany. I came to Bunce Court in 1939, and remained there until 1948.

In 1948 I started college, first at Imperial College, London University and then at Cambridge. At Cambridge University my supervisor was Maurice Wilkes, a pioneer of computer science who is still alive; you might want to interview him. He is now 96 years old, but still as sharp as ever.

ENSMENGER: At that time Wilkes was in the mathematics department, is that correct?

WEGNER: Wilkes was the leader of the computing group within the mathematics department, yes. I don't believe at that point computer science had its own department. But Wilkes designed the first British computer. He was knighted in the year 2000 for that accomplishment.

ENSMENGER: Did you go to Cambridge intending to work with Wilkes, or was that coincidental?

WEGNER: I went to Cambridge because I was invited by Douglas Hartree. We had met when he gave a talk at Imperial College, and he offered me a summer job programming for the EDSAC after I completed my undergraduate degree.

ENSMENGER: Yes.

WEGNER: Oh, you recognize the name?

ENSMENGER: Yes.

WEGNER: Hartree was the person who greeted me on arrival in Cambridge. Ultimately, however, I ended up working mostly with Maurice Wilkes, who was then head of the department. I finished the summer course and then completed the equivalent of a master's degree. This was in 1954 -- which may make me the first person ever to receive a post-graduate diploma in computing. It was awarded by what was then called the Department of Numerical Analysis and Automatic Computing. You asked me earlier about the connection between mathematics and computer science, and of course numerical analysis was what mathematicians thought was the most important aspect of computing.

ENSMENGER: And you were at Cambridge from roughly 1949 to 1954?

WEGNER: No. I was mainly at London University. I was at Cambridge for only a year -- from 1953-54. In 1953 I received my degree from London University. Interestingly, there is another faculty member here at Brown who likewise got his degree at Imperial College. He is now a member of the British Royal Society. We occasionally interact in our professional capacities.

It was at Cambridge that I met my wife Judith; she was at Newnham College reading law, and we met during a dinner at the Cambridge University Jewish Society. Just one of the many wonderful encounters I experienced at Cambridge...

ENSMENGER: And your degree at London University: what department was it from?

WEGNER: You might say mathematics. It was indeed technically an undergraduate degree in mathematics at Imperial College, but I also studied philosophy at University College with Professor A. J. Ayer. I have recently written a little about Ayer, who instilled in me a life-long interest in philosophy.

So that's my background. I later got a PhD degree to complement my diploma from Cambridge. After working for a time at Manchester University (programming on the Atlas computer), and spending some time in Israel working at the Weizmann Institute, I returned to England in 1955.

In July 1956, Judith and I got married, and a year later we left for the United States. I worked first at Penn State, and then spent a year at MIT and a year at Harvard. Later, I spent three years at Cornell University, when Cornell started its computer science department. And finally, I moved to Brown in 1969.

ENSMENGER: You alluded to the fact that what we today call computer science has been called different things, and your career began at a time when was still some negotiation about what that field will be called. In fact I was reading a paper of yours from 1970 and you still referred to it as “computology,” a term that people were using at the time. I’m just wondering when you were at Penn State, for example, what department were you in?

WEGNER: I was in the engineering department. My job was to assist other researchers in using the computer. So I was in effect the engineering department’s computer guru, although the computer I worked on was really the only—or at least the main—computer at Penn State at the time.

ENSMENGER: Right. Do you happen to remember what computer it was, specifically?

WEGNER: It was called the PENNSTAC -- not a well-known computer. In Cambridge, I had worked on the EDSAC, the first true stored-program computer. Invented, of course, in England! Maurice Wilkes, who built it, beat out Alan Turing, who wanted to build his own machine but never actually did so. Later we will have to talk about Turing: he was obviously a very important figure in the history of computing, but I think his role has been much misinterpreted.

ENSMENGER: So, as we talk about that later, a question I will have is whether you think the rediscovery of Turing is a modern phenomenon—whether Turing in a sense has been re-invented by modern computer scientists. Do you remember Turing being as prominent when you were studying in the 1950s?

WEGNER: Actually, it was in June 1954, just as I was finishing up my Cambridge diploma, that Turing killed himself. Because he had been a Cambridge professor, my advisor Maurice Wilkes knew him quite well. They in fact disagreed about many things, but Wilkes was saddened when he died. That is how I first found out about him.

I think that Turing was central in the evolution of computer science, particularly in the 1960s, which is when computer science started to become a substantive discipline. That’s actually one of the big questions: when did computer science become a substantive discipline? My own opinion is that it was in the mid-1960s; if you want an actual date, in the years 1964 and 1965.

ENSMENGER: Is there a particular event or conference or publication in this period that you see as a turning point?

WEGNER: Yes. That’s a good question. And I happen to have an answer that I was going to give you anyway! It was in those years that the ACM convened its first curriculum committee. This was the committee that published the famous Curriculum ’68 report a few years later. And I just happen to have been the person who wrote most of that document.

ENSMENGER: Ah, that is a question I did have for you.

WEGNER: So I was personally involved in actually writing the report, although of course I did work with others as well. But the 1968 curriculum report is the first substantive report that had a detailed report of what a computer scientist program should teach.

ENSMENGER: Let's perhaps develop that story a little. I was reading some of your earlier papers over the past week, and one of them referred to something that I believe was called the ACM College Consulting service, or it was obviously part of an ACM committee that was visiting, in this paper, small colleges discussing curriculum issues. Could you talk a little bit about how you first got started with the ACM curricular activities? How did that come about? Who was driving that particular movement?

WEGNER: You know the famous book by Saul Rosen...

ENSMENGER: Yes, I have it on my shelf.

WEGNER: Saul was an important person in the development of that curriculum. But the person in charge was Bill Acheson of the University of Maryland, who was later replaced by Sam Conte of Purdue University. But I ended up being the one who actually wrote the 1968 Curriculum Committee report, with the assistance of Earl Schweppe, the Committee secretary.

ENSMENGER: And the Curriculum '68, that was actually published in '68? Or was it published slightly earlier?

WEGNER: Curriculum '68 was published in 1968 in the *CACM*. I don't have a copy with me now, although I did look at it several times in the last few months.

ENSMENGER: I have a copy on my computer, if you want to refer to it later.

WEGNER: So anyway, I was the person who actually wrote the report, though I did write it in agreement with all the people on the committee.

ENSMENGER: And just to be clear, this is a process that for you began around 1965 and culminated in the 1968 report.

WEGNER: There is also a 1965 curriculum statement, although it was not actually a Curriculum Committee report. But basically it said that the Curriculum Committee was meeting and was reviewing such and such issues and was in the process of preparing a report. That statement was also published in the *CACM*—probably in 1965. This was before I actually joined the committee. I joined it in either late '65 or early '66.

ENSMENGER: So my memory of the Curriculum '68 document is that it's attempting to lay out a structure for a computer science curriculum, and it is reconciling, among other things, the theoretical approach, particularly the mathematical, with what you might perhaps call the engineering approach, with its focus on technical programming languages.

WEGNER: Definitely the question of the role of mathematics versus engineering was one of the factors being considered. My own thought at the time was that it was vitally important that we develop a theoretical basis for computing. To that end I interacted with several of the people who were working on theoretical computer science, including Donald Knuth. Knuth and I started meeting sometime in '67 or early '68. Knuth had not yet published Volume One [of his series, *The Art of Programming*], and I was in the process of preparing my own book, *Programming Languages, Information Structures, and Machine Organization (PLISMO)*, which came out in 1969.

ENSMENGER: I also have a copy of this on my shelf.

WEGNER: That book was published in 1969. When Donald Knuth and I talked in '68—I think we were hiking on Mt. Wilson in Los Angeles at the time—we discovered that we shared similar interests. I actually first met Don in Las Vegas at a meeting where there were quite a few publishers—they were meeting with both him and me because we both happened to be working on books. We got along immediately. After the Las Vegas meeting, Don drove me to his house in Los Angeles—this was about a five- or six-hour drive—where I stayed with him. That was when he showed me an outline of the seven books he was planning on writing. You know the seven books?

ENSMENGER: Of course.

WEGNER: Well, he showed me a draft of some material from each of the books. And of course it was just before Book One came out.

ENSMENGER: Right, I was going to say, Volume One gets published in '68, does it not? '68 or '69?

WEGNER: '68 or '69. It may have been '69, or it may have been '68; I'm not entirely sure. But this was before they were published—he was still thinking the whole project through.

ENSMENGER: So you were a key author of this influential Curriculum '68 report, and this report addressed the tension between mathematical foundations versus, say, practical training in programming. Do you remember much of the debate from this period? For example, do you feel as if the Curriculum '68 recommendations, did they cause controversy, were they widely accepted—what happened as a result of that?

WEGNER: I think they were quite widely accepted, but they also caused controversy. The debate about curriculum continued—still continues—to concern people. My committee, for example, included several mathematicians, who of course felt that mathematics was essential to the curriculum. I agreed, but I also thought that it was not all that should be studied. There are other aspects of the discipline.

ENSMENGER: Well, you know what, let's hold that discussion for later, because this period in the 1960s where you're defining what computer science is going to look like I think is very important, and I'd like to develop that. Can we take a step back, and do you happen to remember when you joined the ACM? I know that by 1965 you're involved with the committee. Do you

remember when you first joined the ACM?

WEGNER: One thing I can say is that in 1961 I returned to England from the United States to take up the post of Lecturer at the London School of Economics. But I think I joined the ACM before that. I definitely remember being invited to speak at ACM meetings in 1960 and '61.

ENSMENGER: Which of the three major professional societies in computing in this period -- the ACM, the IEEE Computer Science, and the Data Processing Management Association, the DPMA—did you join?

WEGNER: I joined only the ACM. I may have looked into the other professional societies, but I was never seriously involved in them.

ENSMENGER: Right. And of the three, the ACM had the largest percentage of academics, was the most academically oriented.

WEGNER: That's probably why I chose to be involved in it.

ENSMENGER: The ACM certainly the most mathematically oriented of the three, which would obviously fit into your interests. I was just wondering if that was a deliberate decision on your part as you were thinking about your career and your agenda for the field?

WEGNER: I don't remember whether it was so deliberate, but I clearly saw the ACM as the society I wanted to be a member of. I mentioned earlier that in 1961 I went back to England. Prior to that, however, I was working at MIT and Harvard. At MIT I worked with Fernando Corbato on the Multics project. Multics was sponsored by Bell Labs, and it was while working on Multics that I first became involved with the ACM. Being part of the ACM was the reasonable thing to do during that time.

ENSMENGER: One of your accomplishments, as you mentioned, was to be one of the first people to receive a degree in something that looks like modern computing. [Yes] But then you're in an engineering department at Penn State. At MIT you were at the Multics lab. What department were you in at Harvard when you were there?

WEGNER: Actually at Harvard the department most directly involved with computing was statistics. I did interact somewhat with Howard Aiken, who was still floating around. In mathematics or engineering, I believe...

ENSMENGER: How about when you return to England? You're at the London School, but what department are you in there?

WEGNER: At LSE, I was in the statistics department, in part because that's where I had been at Harvard. I interacted with Karl Popper and Imre Lakatos, participating in their philosophy seminar, and with Maurice Kendall, writing a 10-page study of statistical computing. I organized a conference on system programming, which led to an edited book, *Introduction to System Programming*, for which I earned praise from Donald Knuth when we met in the late 1960's. My computer science course for LSE students was supplemented by an evening seminar

attended by over 100 workers in the field. However, in 1964 I decided to move back to the United States with my family, in part because of the inadequacy of my LSE salary.

ENSMENGER: During this period you're developing a kind of intellectual core for computer science. What kind of a relationship do you have with industry, if any? Are you doing any consulting work, for example?

WEGNER: Yes, I was doing some consulting work. In about 1956 or so I did some work for the Prudential Insurance Company developing actuarial tables. I had actually studied to be an actuary—although I never got around to it—and worked on actuarial computing for the Pru at some point in the mid-1950s.

ENSMENGER: Prudential has, as you well know, an interesting role in the history of the computer industry in the United States. In this same period they are funding some of the work on UNIVAC, and they've got Edmund Berkeley working as a consultant for them. Berkeley, as you know, goes on to write the first popular book about computing.

WEGNER: My first stint with Prudential was limited to using the computer to construct actuarial tables. Not that interesting, really. But when I came back to England to work at the London School of Economics, Prudential tried to get me to come back to do some more significant work..

ENSMENGER: Right. So one of the reasons for asking that question about industry is that I suspect, that just as you are finishing up the Curriculum '68 report, there is increasing pressure on computing departments to produce workers—programmers, systems analysts, other kinds of people who won't have academic careers but will do some programming in that business context. Do you remember having explicit discussions about the role of that kind of technical training in the Curriculum '68 documents?

WEGNER: I'm sure I did, although I can't recall the details. But that was a subject that always interested me, namely, what is the status of computing as an intellectual and professional discipline. What did it mean to study computing? When I started working with computers in the 1950s, there were no careers in computing. Certainly not the remunerative careers that emerged later. I just saw it as an interesting intellectual challenge. This all changed in the 1960s.

People often tell me that my work is unusually philosophy-oriented (at any rate, for a computer scientist). This relates, in some measure, to my personal history as a Holocaust survivor. When I talk to high-school classes about the Holocaust, I encourage them to think about ways in which they can help make the world a better place. I suppose I have always thought about my work in computing as my contribution to making the world a better place. I can't help thinking about the bigger issues.

ENSMENGER: One of the questions I'd like to ask you later is about the way in which you feel computer science has influenced other disciplines. But if we could keep it in the 1960s, can you remember what you thought...did you have a grand vision for computer science influencing other fields, or making the world a better place as you remember it in the 1960s as the field is developing? Do you remember having those aspirations?

WEGNER: Well, I'll tell you what actually happened, and that is that I became more and more involved in programming languages as a discipline. So I worked in the field of programming languages more and more, and less on mathematics or theory. And in the period 1968 to '71, I actually became the head of SIGPLAN, the ACM Special Interests Group in Programming Languages. I was in fact the very first chairman of SIG PLAN, and I helped develop the discipline of programming languages during that period. This was a period in which programming languages were a huge priority for the computing community. Nowadays programming languages are not seen as being so central, but at the time this was an important area of research. Look at how many Turing Award winners are programming-language people. Alan Perlis, for example. He and I were friends right up to his death in 1990.

ENSMENGER: Now if I remember correctly, the Curriculum '68 document is language-agnostic. It doesn't spend a lot of time talking about specific programming languages. Your PLISMO book does, however.

WEGNER: PLISMO was published after the Curriculum '68 report, of course.

ENSMENGER: Right. So in between the two...

WEGNER: The book doesn't really focus on a single language, but rather talks about languages in general. Did you mean, did I have some particular concern with a specific language? ALGOL 60, for example? I don't think so. I was primarily concerned with models of computing for all languages.

ENSMENGER: Well, since you mention it, one of the questions I was going to ask was about ALGOL. In this period there are discussions about which programming languages would be most appropriate to build into a computer science curriculum. ALGOL seems to be, for a time at least, the language of choice of the ACM. I was wondering what you remember about those discussions.

WEGNER: Well, I was always interested in ALGOL, and I wrote several short articles about its properties, which were published by Peter Naur in the ALGOL Bulletin. But I was never fanatical about ALGOL, though some of my colleagues were. .

ENSMENGER: Were you involved in any of the language standards committees in this period?

WEGNER: No, I was not. But I interacted with people on the language standards committees. I interacted quite a bit with McCarthy about LISP. In 1959-60, when I was at MIT, I worked in the room next to McCarthy and got to know him well.

ENSMENGER: So, again, if I am remembering correctly about this period, there's a discussion in programming languages about formal proofs and about structured programming. And I know you worked with Maurice Wilkes. And among the things I love about Wilkes are his stories about programming the EDSAC, and particularly about discovering bugs and the process of debugging. And it seemed to me that he always had a very good sense of the balance between the theoretical aspects of computing and the practical aspects of writing programs. It seems to

me that that's something you share with him -- or inherited from him.

WEGNER: I would say I shared it with him, but again I would say that he was never a theorist. He certainly was interested in both theory and practice, but you could never have called him a theorist as such. He was, if anything, more practical than theoretical. But he certainly was interested in the role of theory and the role of practice, and in that respect we agreed. We still meet occasionally. We last met about 18 months ago in Cambridge, England. And back in the 1980's Wilkes was working for some years at DEC in Maynard, Mass., so we saw each other then. We used to meet once or twice a year. I'd have him give lectures here at Brown, and I occasionally drove up to have lunch with him.

ENSMENGER: I was reading a 1970 article by you called "The Three Cultures of Computing".

WEGNER: Yes, yes. That was a good article.

ENSMENGER: It was. It's a wonderful article, and much referenced by historians of computing. You describe the three cultures of computer technology, computer mathematics, and computer science. I noticed that already by this period you were talking about computer science research in terms of paradigms—obviously a reflection of your philosophical bent.

WEGNER: Yes

ENSMENGER: And I was surprised at how early on you were using that language...

WEGNER: Of the paradigm, yes.

ENSMENGER: ...and reflecting back on the field. Maybe this is a slightly different question that will get at the same basic concept, which is if you think back to these discussions around Curriculum '68. Were you pleased with the immediate aftermath of Curriculum '68? Did you think that the field developed along the lines that you had hoped it would when you wrote that report?

WEGNER: I think my answer would actually be no. I think that on the whole, people are not as thoughtful about developing a substantial theoretical basis for computer science as I would like them to be. I still go to faculty meetings nowadays, and I sometimes bring up what should be the nature of computer science. They always move away from that. They're never interested in those kind of conversations. I have suggested that we collectively write an article about what we think computer science actually is (or should be). They don't actually say no, but we never seem to get around to it.

ENSMENGER: Right. We talked earlier about Peter Denning and he's also written quite a lot recently about this question.

WEGNER: Yes, yes. Now, Peter Denning, I read his stuff and interact with him and comment on his work. Peter Denning is leading the pack in this regard. And of course he is also a former president of the ACM, so he has the respect of the community.

I think I first met Peter while serving on the ACM Council from '72 to '76. Peter became president of the ACM probably around '76, something like that.

ENSMENGER: Right. Can you, for those of us who aren't familiar, talk a little bit about the role of the ACM Council at that period?

WEGNER: Well, okay. It was the primary high-level committee that decided where the ACM was going and what it should focus on. The president usually participated in this discussion, along with various other leaders of the field. It was a high-level committee. Very much focused on the big picture.

ENSMENGER: Do you remember any particularly contentious debates from your time on the Council?

WEGNER: Not well, no. No. I made a point of attending the meetings, and I enjoyed the discussions, but I don't recall that we ever accomplished anything very important.

ENSMENGER: Now you just handed me an article of yours from 1972 on education, which I happen to have read just last night.

WEGNER: Oh, you actually saw this particular article?

ENSMENGER: Yes, this specific article. And you'll see in a minute why I remember it. But this was published after Curriculum '68, during a period when you published quite a bit on graduate education and various other curricular issues. You say something in this article, I believe, about the role of the PhD in computer science. You talk about it in terms of professionalization, about how people were getting PhDs in computer science for professional reasons but not for intellectual reasons. It's something of a critique of computer science as a discipline.

WEGNER: Well, at the time I thought that many of the people going into computer science were more interested in what they could use it for—industrial and commercial applications—than in really doing something that would help both the discipline and the world as a whole. It comes back to this idea that we should all be working to make the world a better place.

ENSMENGER: Do you feel that that was true of the computer science community in the 1970s? Do you believe that that's still true? Is this something that's changed over time for you?

WEGNER: Yes, I believe that it's still true.

ENSMENGER: So I was asking if this perception of the field was true only in this period of the 1970s when you're writing a lot about curriculum and education, or if you still felt that the field was insufficiently theoretical.

WEGNER: Actually I thought sometimes that it was overly theoretical. You know, I thought it was insufficiently understood. Even today this is true. There are people like Moshe Vardi, who is now editor of the Communications of the ACM, and Peter Denning, who are doing a good job

addressing these issues. Vardi, for example, has been critical of the narrow focus of the peer review process, which too often reinforces conventional wisdom and discourages big-picture thinking.

ENSMENGER: Do you believe that's a structural problem with peer review generally in the sciences or something unique to computer science as a discipline?

WEGNER: I think it's a more general problem with peer review. But it is definitely a problem in computer science.

ENSMENGER: Tell me a little about how you are addressing this problem in your most recent book, which is called *Interactive Computation: The New Paradigm*.

WEGNER: Interactive computing is a topic that has interested me during the last twenty years. I am not sure that it is necessarily the new paradigm that will change everything—although many of the contributors to the volume think so—but I believe it is an important direction for the field to pursue, because it extends algorithms to broader interactive problem solving.

ENSMENGER: This is a book you've edited with Dina Goldin and Scott Smolka, and it was published in 2006, correct?

WEGNER: Yes, it is a recent book, written late in my life. The contributing authors are excellent: Farhad Arbab, Robin Milner, Manfred Broy, Moshe Vardi, Orna Kupferman, Jan van Leeuwen, Yuri Gurevich. Great people. All of these authors are really quite interesting. There is also a paper by me and Dina, and an introduction that reviews all contributions.

ENSMENGER: So this is a book that reintroduces the language of the paradigm—something that I've heard you speak about in other contexts. And you obviously aren't using it in a casual sense: you've read your Thomas Kuhn and use it in a very technical, philosophical sense.

WEGNER: Yes—although I recently had an argument with a faculty colleague who said that I talk too much about paradigms, that I should throw out paradigms, that people are interested in other things more important than paradigms. But I think it is still a useful concept in describing conceptual changes..

ENSMENGER: In the language of the history and philosophy of science, paradigm has a very technical meaning. It also has a more casual meaning—people in the popular media are always talking about paradigm shifts. But when Kuhn talks about paradigms he's talking about a combination of ideas and practices, and particularly about the problem of moving from one paradigm of thinking to another.

WEGNER: Yes, yes. Of course I've read Kuhn quite carefully, although I don't always agree with him. Sometimes people say of Kuhn, "But he really doesn't understand what a paradigm is." But it is not so much that he doesn't understand what a paradigm is, just that he is rather too loose in defining the term, and he uses it in several contradictory ways.

ENSMENGER: I agree.

WEGNER: Let me tell you something about what I have been doing recently. I'm the editor of a Brown University journal, *The Brown Faculty Bulletin*. This is the issue that came out about two weeks ago. Notice that there's an article by me about Isaac Newton and Stephen Hawking. If you know who Isaac Newton is, you probably know who Stephen Hawking is. Stephen Hawking is the Lucasian Professor in Cambridge -- the same position Isaac Newton held. In any case, the article is about how paradigm shifts in physics have occurred, and talks about the ways in which Newton resisted change in his field. He was not always a nice guy, though he was a good researcher.

ENSMENGER: But it suggests the ways in which in a Kuhnian sense paradigms come to define what a discipline is about, and that shifting what a discipline is about can be quite difficult.

WEGNER: Oh, yes. That is true. I agree with you.

ENSMENGER: It might be interesting for us to talk about shifts that you have seen happen in the discipline, and what those tensions were and how they played out in practice. Let me just follow up briefly on a point you made earlier about your role in the ACM publication committee. You were the editor and chief of the ACM press, if I have this date correct, from 1987 to 1992, and of the ACM computing surveys from 1995 to 1999.

WEGNER: Okay, yes. Those are the two extra things that I haven't yet mentioned, that I wanted to mention about my work with the ACM, yes.

ENSMENGER: Okay. And you can tell me whatever you would like about that. Obviously as an editor, you have a powerful role in shaping the discipline. So that might be a question to start with, which is what did you hope to accomplish during that period? Do you feel as if you were able to accomplish it?

WEGNER: When they invited me to serve as editor-in-chief, I was of course pleased that they thought so highly of me and my work, and I wanted to work with the ACM. That's another way of trying to make the world a better place. I worked hard in my role as editor to pursue the general interest of the field rather than my own individual agenda. I tried to hire editorial assistants that I thought would be good and fair and reasonable. I think I did an okay job. I was a good editor—not great, but good.

ENSMENGER: Did you enjoy that role?

WEGNER: Yes, certainly. I wouldn't have done it if I didn't enjoy it. I also think I had a broader understanding of the field than many people—I was always interested in trying to understand the field in many different areas, many different ways—which helped me do a good job.

ENSMENGER: Your mention of the breadth of your experience reminded me that, chronologically speaking, we've jumped a bit in time. The last that we left you, you were at the London School of Economics. Perhaps you could just tell us a little bit about what happened immediately following that.

WEGNER: Well I was at the London School from '61-'64. In '63 I was invited to the States to Bell Labs to spend the summer at Bell Labs. And the person who invited me, with whom I did some of my work, was a guy called Douglas McIlroy. Do you know him?

ENSMENGER: Yes, Doug McIlroy. Yes. Not personally, but I know his work very well.

WEGNER: He was the one who invited me to Bell Labs. I had also, by the way, worked with him at MIT.

ENSMENGER: I've read quite a bit of McIlroy in the context of the 1968 NATO conferences on software engineering.

WEGNER: Yes.

ENSMENGER: He's quite a prominent voice in that discussion.

WEGNER: Yes—although I knew him best around 1963, which is prior to that conference. And I didn't keep up with him very much after that.

ENSMENGER: So in '63 you're invited to Bell Labs, he's a part of that, is that correct?

WEGNER: Yes. He was in a sense the main guy. I believe he was my supervisor when I was there. Incidentally, one of the things that I learned at Bell Labs in 1963 was the game of Go.

ENSMENGER: Yes, Go. I know the game.

WEGNER: So I do play Go quite well. And I've also visited Tokyo occasionally. On one occasion when I was there, I was asked to play Go against a computer. They were trying to make the computer play Go as well as a person. And it was a big deal. It took at least five hours. There were 20 or 30 people watching me play. At one point, they asked the computer "How is he doing compared to you?" and it said, "We're beating him." But it turns out that I beat that thing by a large margin of over 100 points.

ENSMENGER: I've been doing some research lately on chess in computing, and obviously part of that discussion is why Go is so much more difficult a problem for a computer to solve than chess.

WEGNER: With regard to chess, I actually was invited to watch the chess tournament where Deep Blue finally beat Gary Kasparov. I got the invitation because the daughter of the guy who programmed the computer for chess was taking my class at Brown. I didn't personally have anything to do with it, but it was fun to watch the computer beat the world champion.

ENSMENGER: Not to take us too far afield, but since we're talking about that moment [when Deep Blue beat Kasparov], do you think that was a significant moment for computing, or that it was just an exciting event that got a lot of publicity?

WEGNER: I think it was an exciting event, but it wasn't hugely significant. I think AI is an interesting discipline, but I am not sure computer chess is all that significant anymore.

[Part 2]

ENSMENGER: Okay, when we broke for technical reasons, we were talking about your arrival at Brown University, and about the development over time of that department. You spent your first ten years at Brown in applied mathematics, and in 1979 established the Department of Computer Science. If you had to characterize the changes that have happened in the department relative to the larger changes in computer science as a discipline over the past several decades, do you see Brown as reflecting those larger changes? Does the Brown department have a unique identity?

WEGNER: When we first started the department we had about twelve faculty. Now we have over twenty. We've grown, obviously. And we do have more or less the same subjects we had when I arrived—with the possible exception of new work in computational biology.

ENSMENGER: Many of the people who work in the computer industry developing software do not have formal training in computer science. Do you see that as a problem, as a failure of the discipline?

WEGNER: Well, I don't think of it as a failure of the field, but I do think that if you don't have training in computer science you're at a disadvantage—particularly when dealing with more complicated software problems.

ENSMENGER: What do you think are the key texts in computer science? What books have you found most influential in your own thinking? We talked about Knuth's work, for example. Are there other books that you think have been historically significant or that you continue to think are central texts in the discipline?

WEGNER: I can tell you about texts that have been important in my reading and research. These include Donald Knuth's writings and the books by Jeff Ullman about the theory of computation and data bases. Also books by Zohar Manna and David Harel in the 1980s on the theory of computation.

During the past year, I have read Bertrand Russell's *History of Western Philosophy*, Thomas Kuhn's *The Structure of Scientific Revolutions*, and John Horgan's *The End of Science* (which describes the views of authors like Popper, Penrose, Kuhn, Hawking and other well-known scientists). I am currently interested in the historical and philosophical evolution of science, with a special focus on computer science.

ENSMENGER: So you obviously read broadly in philosophy and history. What do you think about the way computer science as a discipline engages with other intellectual pursuits? And I don't mean mathematics or engineering. I mean philosophy, history, social sciences—those kinds of things.

WEGNER: Well, I certainly think interaction with other disciplines is important. I am not sure

the larger computer science community does. I believe every researcher has his or her own view.

ENSMENGER: We've talked about a lot about as a discipline; we've talked a little bit about your work, limited work, in industry. Could we talk briefly before we break just about education? Do you think that computer science programs are doing a good job educating students?

WEGNER: Well, perhaps because I am getting older and no longer teach, I don't think about education as something that I have much to contribute to anymore. I do think that on the whole people are not as interested as they might be in addressing the implications of their work for broader social and political questions. Many of our graduates are more concerned with things like making money, acquiring status, and so on. And they're not sufficiently interested in how the community can help the world become a better place. In that respect, I think our educational system could be improved.

ENSMENGER: We haven't talked much about your own research specialty. We've talked about the structure of the field generally and about the computer science program here at Brown. Perhaps we could return to your very earliest research and talk about what you thought your research was going to look like as you began it in the '50s and '60s and how it's changed over time.

WEGNER: Well, much of my research has been concerned with synthesis. I am good at understanding and evaluating the work of others and presenting that in a clear and coherent form. I haven't invented any new programming languages, for example, although I have written a great deal about the theory and practice of programming languages. Since the 1980s I have been much interested in object-oriented programming. The work of Barbara Liskov, for example, who recently won a Turing Award.

ENSMENGER: Now you have written quite a bit about ALGOL. ALGOL was a language that in the United States, at least, never became widely popular in the industry. Do you have a sense of why ALGOL didn't do what people had expected of it? Or why it took off differently in Europe compared to the United States?

WEGNER: Yes, and even in Europe it did not become a widely used language, but it became a language that was more interesting for theoretical purposes; people liked it more as a model than as a language that could solve real problems.

ENSMENGER: So on a conceptual or theoretical level it was appealing but in practical terms...?

WEGNER: In practical terms it ended up not doing as well as it planned. Partly because its capacity for doing input/output was deeply flawed. And a bunch of other things. It had some good ideas but not a complete set of tools for doing all the things that a programming language needs to do. A lot of languages are like this. When all is said and done, I think computer science is at a stage where we're still trying to design new programming languages, yet don't have a method of producing a complete model, so these efforts generally don't work out.

ENSMENGER: Do you think that that is a problem of just not having created the model yet, or do you think it's a situation in which there is no model?

WEGNER: Well, I think both of those explanations are equally valid. It depends on what you mean by there being a model —what does it mean? I think creating a complete model is difficult, and there are certain things that are more difficult than people realize. For example, all of the attempts to develop a complete model for mathematics have failed—are doomed to failure. That's not quite the same as a model of a programming language, but it's not that far removed. Making a complete model of something or other is far more difficult than people imagine.

ENSMENGER: At what level do you think that mathematics is essential to computer programming? Is it at the level of operating systems, at the level of language design, or at lower levels of application?

WEGNER: I think that mathematics is sometimes useful but its usefulness is often overestimated. So I find that when people think mathematics is the primary way of including computational techniques or evaluating them, they are generally wrong. One example of that is the case of Alan Turing. Turing was originally a mathematician, and his Turing Machine, although an interesting concept, has ended up not being a useful general-purpose technique for computing. Many people in the late 1960s and 1970s tried to adapt Turing Machines to be a computer model, but they were unsuccessful, and one of the things they asserted was that it was a mathematical model and that the foundation of the field should be formal and mathematical. And there they were wrong --- or at least unsuccessful. So I think the characterization of the Turing Machine as a complete model for everything was overly optimistic. I have written some papers about this, which were strongly criticized by the people who believe Turing machines constitute a complete model.

ENSMENGER: What do you think of the idea that the Universal Turing Machine is the fundamental unit of analysis of computer science? Why is that so appealing? Is it the idea that it could potentially be mathematically formalized? Why do you think that idea took hold?

WEGNER: At the high level, the appeal is that people are trying to develop an abstract, generalizable mathematical theory of everything. This goes back to Pythagoras. The Newtonian revolution placed mathematics at the center of physics, and other scientific disciplines followed suit. The 20th century challenged some of this faith in mathematics — consider the work of Godel and Turing, for example --- but computing followed this basic trend. Mathematics was the answer; mathematics could provide a solid foundation for the discipline. I think mathematics is important --- obviously --- but it is also insufficient.

Another problem was the search for other kinds of formalism, such as formal procedures --- having a model where no matter what the problem was, it had a complete solution. That approach also didn't work out.

The problem is that there are many applications where what you're trying to solve is not entirely mathematical. Some problems are mathematical --- which works for the model --- but there remain plenty of non-mathematical aspects. In this case, these formal systems proved over-ambitious. People want complete and systematic methods for solving every problem, but that is

not how the real world works.

ENSMENGER: As I understand your recent research, you've been arguing against some of these earlier systematic models and paradigms and have in fact suggested a new paradigm, the subtitle of your most recent book. Can you briefly describe your own thinking about current paradigms of computer science?

WEGNER: Well, a new paradigm may not be a complete paradigm, but it is one that describes things in a way that previous paradigms could not. It depends on what you mean by a new paradigm. It is new, and it is a paradigm for certain kinds of models, but by no means for all models.

ENSMENGER: So I was borrowing that from your new book on interactive computation, of which the subtitle is "A New Paradigm".

WEGNER: This work began with an article titled "The Paradigm Shift From Algorithm to Interaction," published in the CACM in May 1997. Let me read the opening paragraph:

"The paradigm shift from algorithm to interaction captures the technology shift from mainframes to workstations and networks from number crunching to imbedded systems and graphical user interfaces, and from procedure oriented to object oriented and distributed programming. Interaction is shown to be more powerful than rule-based algorithms for computer problem solving, overturning the prevalent view that all computing is expressible as algorithms. The radical notion that interactive systems are more powerful problem solving engines than algorithms is the basis for a new paradigm for computing technology built around the unifying concept of interaction."

Okay, so for a long while algorithms were considered to be the essence of computing -- an idea supported by Knuth in his first book...

ENSMENGER: The subtitle of which was "Fundamental Algorithms," if I remember correctly.

WEGNER: Yes. But in a recent discussion, Knuth agreed that there was more to computing than algorithms. So what we are saying in this new book is that algorithms do not provide a complete model for problem solving. That's a substantive assertion because algorithms were for a long time considered to be a complete model. Just as Turing Machines were seen as a complete model, so algorithms were likewise considered to be a complete model for problem solving. And so the contributors to this new volume are attempting to move beyond algorithms.

ENSMENGER: So if we were to ask the question that was asked in the 1960s as computer science was coming into being, what is computer science the science of? There have been various attempts. Herbert Simon argues that a computer in itself was worthy of a science. As you suggest, Knuth argues (or at least argued) that it's the algorithm that's at the core. Others have argued that it's the Turing Machine. You're challenging, I think, all those different conceptions of what computer science is the science of. What then is your own current answer to that question?

WEGNER: Well, my answer is that we cannot answer the question, "What is computer science?" but we can add some ideas beyond the things that we've so far discussed. And the fact that we can add ideas means that thus far we've not completely described it. So my answer is that complete description is intrinsically impossible. And I think I've said something equivalent to that several times today -- that we can't do it for mathematics, we can't do it for physics, and we can't do it for computer science. We can't have a complete answer. But we can find new ways of thinking about parts of the answer.

ENSMENGER: Right. And in that sense, one of the challenges made to computer science in the '60s is that if you didn't have a comprehensive unifying theory, it wasn't a science. But you're arguing that physics doesn't have that, and we still have physics and the study of physical systems is still useful, and we don't have that in mathematics, and yet the study of mathematics is still useful and productive.

WEGNER: Yes.

ENSMENGER: Are you saying something similar about computer science?

WEGNER: Absolutely.

ENSMENGER: Despite the fact that there's no unifying theory, that's not an argument against the validity of computer science as a discipline?

WEGNER: Though incomplete, it is useful in productive ways. And this is also true of software engineering—where we don't have a complete description but it is still of practical use.

ENSMENGER: Right, a useful way of thinking.

WEGNER: And we have to do that. And as I have said elsewhere, the same problem arises in politics and in religion. Similar principles are involved. So this whole notion, which has been used very widely, is not a good notion. We cannot be complete. The completeness goes beyond in many different areas.

ENSMENGER: So your new research on interactivity is a new paradigm in the sense that it's adding to these layers [yes], but not a new paradigm in the sense that it's replacing the other paradigms.

WEGNER: It's not a complete thing, but it's an extra layer.

ENSMENGER: So this 1997 publication is the first of your papers on this broad topic of interaction that culminates in the 2006 volume, is that correct?

WEGNER: Not exactly the first, but the first important paper. There are other earlier papers as well.

ENSMENGER: As I'm looking around your office, quite visible is your distinguished service award that you received from the ACM in the year 2000, and I assume that this is in recognition

of your long participation on various committees, including the ACM Press Editorial Board. What is the story behind this particular award?

WEGNER: Well, I was pleased to receive it, of course. I wasn't given a Turing Award -- my work generally is not on the theoretical level necessary to receive that particular prize --- but the ACM award captures one of the things that I think I do best, which is to synthesize cutting-edge research and help make it more broadly accessible.

ENSMENGER: We are coming to the close of our session, but is there anything you would like to say about recent developments in computers? What about the personal computer and the Internet, for example? How do you think they have influenced the field?

WEGNER: I think these technologies have changed the way people perform research, but I'm not sure this provides a good general model for revolutionizing research. But they did change the way people did research.

ENSMENGER: You mean on the practical level of writing papers and developing systems, or that it was a new kind of machine for them to think about?

WEGNER: On the practical level, yes. It changed the way people did research, but it still remained the case that software technologies didn't always give good answers, being incomplete in their problem-solving ability. Even though we now have better forms of computer specification, there are still no uniformly good methods of complete problem-solving. Though CACM writers like Peter Denning have repeatedly tried to define the profession of IT, no one has yet succeeded in specifying an acceptable paradigm for computing.

ENSMENGER: We have unfortunately reached the end of this session. Thank you so much for taking the time. I enjoyed our conversation immensely.

WEGNER: You're welcome. I enjoyed it too. And I look forward to using the published version of this conversation in my projected autobiography, which I plan to write in the years to come.