An Interview with

CARL K. CHANG, Ph.D.

Interview #

Conducted by Jeffrey R. Yost, Ph.D.

On

23 August 2013

Iowa State University, Ames Iowa

Abstract: Past Computer Society president Carl Chang briefly discusses his early education, graduate education, and the start of his professional career (first at Bell Labs and then University of Illinois Chicago), prior to the bulk of the interview on his many contributions and various leadership roles for the IEEE Computer Society. This includes extensive discussion of COMPSAC, a highly influential and large software and applications conference of the society he came to lead. He also discusses his editorial leadership roles including as EIC of *IEEE Software*, and *IEEE Computer*, as well as various initiatives and contexts to his year as Computer Society President. Chang also discusses the Iowa State University Computer Science Department, where he served as chair for many years, and his evolving research interests.

Yost: My name is Jeffrey Yost from the University of Minnesota and I'm here today with Carl Chang at Iowa State University in Ames, Iowa. This interview is for the IEEE Computer Society Leaders Oral History Initiative. It is the morning of August 23, 2013. Carl, can you begin by just telling me a little bit of biographical information, where you were born, where you grew up?

Chang: Yes, thank you Jeff, for coming all the way to Iowa State. It's a real nice town, Ames, Iowa, in the middle of nowhere, by the cornfields. [Laughs.] My family went to Taiwan from China in 1949 when the Communists took over China. I was born in Taiwan, actually. Taiwan at the time was poor, after World War II; the only thing they had at the time that was still good, I think, is agriculture. My father was a military officer in China, and his major rank was converted to a police office. So we grew up in this kind of family, not the most income. The government was almost broke at the time, but over the course, over the years we see that miracle in Taiwan that resulted in a strong economy. So I kind of, in my childhood, actually went through, witnessed how Taiwan emerged from almost nothing to become a very powerful economy in the world.

Yost: And what year were you born?

Chang: I was born in 1952, so you guess my age, then. [Laughs.] So I was born in a city called Taichung. Taichung is a; chung means middle, middle part. So that's on the west coast of Taiwan, in the middle part of it. And then my family moved to, in 1984, moved to the east coast. That is the northern quarter of the east coast of Taiwan. And then we moved three more times; finally, we settled in Taipei. Pei, of Taipei, means north. So from the middle part of Taiwan, finally we settled in the northern part of Taiwan and Taipei is capital of Taiwan.

Yost: In school, so pre-college, were there subjects you were especially interested in or had a special aptitude for?

Chang: Yes, I can recall that I've been always good in math. I always finished my math homework first in the class. Kind of easy for me. I was pretty good in English, back to those junior high days; the entire class was actually very, very good in terms of performance in English on tests. And then I went to Jian Kuo, Jian Kuo Senior High. In those days, you had to pass the entrance examination and Jian Kuo is the best in Taipei. So I was in their high school for three years. I began to need to decide, at the time, which direction I needed to go for my career. So I decided to go into science and technology, okay, so that's one category. And then I need to pass another entrance examination to get into college.

Yost: And you went to the National University?

Chang: Right. National Central University is located outside Taipei, about 35-40 miles south of Taipei. I majored in mathematics. I thought I had always been good in math and that may be the major to take, and I got my degree. But I think in those days, I enjoyed more the applied math part, which is more than theoretical math. That's how I actually got in touch with computers and computer programming. So I think after the freshman year, I decided I would like to try a degree in computer programming. At the time, Taiwan is still in the early stage of computing. So, I went to National Taiwan University located in Taipei. They offered some courses in computer programming. So FORTRAN became my first programming language, and I learned computer science from there.

Yost: While in college did you learn other programming languages as well?

Chang: FORTRAN and COBOL were the first two languages I learned then, they were popular in Taiwan at the time.

Yost: You graduated in 1974, is that right?

Chang: Right, 1974 I finished the college's degree in mathematics. I was pretty determined to study abroad. But before that, every boy, every young man in Taiwan has to serve in the military. This service is compulsory in Taiwan. So I spend another almost two years in the military service after college degree. And I spent about a year and a half in Kinmen, which is a very small island outside of Taiwan; very, very close to the China mainland. So I spent a year and a half there doing almost nothing, just waiting for my retiring from the military so I could apply to study abroad.

Yost: Then you came to the U.S. to study at Northern Illinois?

Chang: Right. I applied to a few schools; I decided to go to NIU, Northern Illinois University, because it was very close by to where my sister lived at the time. So I spent my first year and a half in Northern Illinois; finished a degree in computer science, my MS degree. I studied a lot of IBM stuff, because Northern Illinois University, at the time, the curriculum was very actively centered with IBM systems. So I learned how to do IBM system programming in assembly language, plus database work. So I think I learned a lot about computer science but still it was limited to a degree because it was IBM centered. So after that, I finished the degree at NIU, I went to GTE. I worked in Northlake, Illinois for the company called GTE Automatic Electric, and I became a systems programmer doing lots of assembly language types of things.

Yost: And how long were you there?

Chang: I spent a year working for GTE. At the time I was thinking maybe I should get more education in computer science. I began to realize it's a broad area, a big field. There are still lots of things to learn and I really wanted to continue my education. So I checked around; I decided Northwestern was not too far; so I went to Northwestern one cold winter. I talked to my advisor, the computer scientist who later became my advisor, Professor Stephen Yau. He was the chairman of the computer science department there, at the time. So I went to see him. I said, I'm now working in industry, but I really want to come back to continue my education, would you take me? He said yes. And that's how I

ended up at Northwestern University. This is 1979. So I started actually in the summer, June 1979, I started my Ph.D. studies.

Yost: Were you at Bell Labs for a period or was that later?

Chang: Bell Lab is after my Ph.D. degree. So I spent about two years and nine months of study to get my degree, which is pretty efficient. [Laughs.]

Yost: Very. Can you tell me about; I understand that your dissertation involved tools or models for software modification for maintenance and updates. Can you elaborate on that a bit and talk about how you came to that particular topic and explain the research briefly?

Chang: Right. Professor Yau, he had several grants at the time from RADC, Rome Air Defense Center; and from Naval Research, and also from the Air Force. I think he has several grants from the military. The part of the project I was involved in was the RADC project, which involved program analysis. So that's how I started in analysis of computer programs. I looked around in Illinois at the end of 1970s, early 1980s; it's an area where lots of people are researching into incremental program analysis or Syntax Directed Editing. So, you have an editor, you don't need to make any syntax error because the system itself knows about the syntax of a particular language. So I look in this sort of area. I decided to call my research Incremental Program Analysis. So we analyzed a program, we produce intermediate code, instead of the binary code or the machine code. So this kind of intermediate representation I had a name for it, it's called a typed tree representation or TTR. So I used that as a foundation to represent programs at the intermediate level, and right down here to set to allow you to make changes to a certain part of program, you only have to analyze that part of the program. So it's the incremental compilation, incremental analysis; decided the area and to finish my thesis.

Yost: Were there any other faculty members at Northwestern besides Professor Yau that were also influential to you?

Chang: Yes indeed. At Northwestern, there's another professor; I met him in Taiwan just a couple weeks ago, again; he's a professor [D. T. Lee]. Professor Lee, when I joined Northwestern University, he was actually a new assistant professor there. He served on my thesis committee. Actually, I also took his course. His course in computer theory, and compiler, and that course opened my eyes. I can really understand a different part of computer science besides programming, that's also formal theory, formal language. So he was very helpful to me, and he served on my thesis committee; and he later left Northwestern University and went back to Taiwan. We still keep close contact. He's now academician in the Academia Sinica, which is Chinese Academy of Sciences' counterpart in Taiwan. Currently he's the president of Chung Hsing University, it's a public university in Taiwan. So I think, of course, Yau, he's definitely my advisor for life, and Professor Lee, he also has had lots of influence on my computer science career.

Yost: You completed your doctorate in 1982.

Chang: That's correct.

Yost: Did you go immediately to Bell Labs?

Chang: I interviewed with Bell Labs; and then I joined the software development support organization; software engineering organization in Bell Labs because of my background in computer programming, also software engineering. So I spent two years there. That also opened my eyes because it gave me first hand information and valuable experience with very large and complex software development. There's I think 700 people working together on one project and how do you support this kind of large scale project; software development project management. So I learned a lot during those two years.

Yost: Can you talk about some of the challenges that you saw in managing large scale software development, large scale software engineering?

Chang: Yes. Bell Labs has a very unique culture, of course. It used to be a monopoly so in Bell Labs it's always a can-do mentality. People, when they miss something, they say oh, maybe a version control issue? No, we can do it. They create their own version control system, their own MR system. MR stands for Modification Request. That means maintenance, maintenance of the evolution of software systems. So I saw how this thing actually works; the inter-working of these kind of things; also outside there, there are version control systems that maintain other things. I can see that the can-do mentality sometimes also brings lots of overhead because you can't do everything by yourself. I think Bell Labs eventually moved away from that; you just cannot do everything by yourself. That also posed lots of management difficulties, complexity, and I learned a lot by observing how they managed a large, complex system. Other than that, I also picked up some knowledge about communication technology, especially telephone switching systems. How do they create a layered structure and architecture; how do you manage different parts of operation and maintenance? This really gave me a solid background, for when I decided to go back to academia. That gave me a very good foundation to go on and teach students about a real, large and complex system; how do you manage such projects.

Yost: When you were at graduate school completing your doctorate, were you thinking more of a career at an industrial research laboratory or an academic career?

Chang: Actually, I was not really thinking about an academic career. I was thinking since my major is software engineering, I will do software engineering. I have to be able to really go through the real work, to understand what engineering really means from the software industry, or software development. So I was pretty determined to go to the industry, at the time until after two years at Bell Lab. One day, I received a phone call from a gentleman, who is Dr. Wai-Kai Chen. Dr. Chen was becoming chair at the University of Illinois-Chicago. The department was called EECS, at the time; Electrical Engineering Computer Science. So he called me; and we actually knew each other before that. He said, would you come here? We have an opening; would you come here for just half a day? You know, half a day is okay, just come here and interview so people can see

you. That gave me the opportunity to go back and explore that and I began to feel like a professor's life actually may not be too bad. So I talked with a few people there and I gave a talk. They liked my talk, because my talk had to do also with down to earth things regarding software engineering and all this stuff about computer programs. So they thought that's a good match. I liked them and enjoyed the visit; that's how I end up going back to academia and started my academic career.

Yost: Software engineering, of course, really gained momentum in the end of the 1960s, and early 1970s . . .

Chang: Yes.

Yost: . . . with the famous NATO, the 1968 conference.

Chang: 1968, right.

Yost: Who did you see as the foremost theorists in software engineering, early in your career?

Chang: Early in my career, of course, we all read Dave Parnas' paper. Frederick Brooks' paper. Those are the pioneers in software engineering. And then I think I also started paying attention to the work of some of these; another camp is the Requirements analysis at requirements engineering group; so Tom DeMarco's work and Ed Yourdon's work. I began to get more and more into trying to understand what is the theory behind software engineering. So those are some of the early names; early work I read as a student, and then as a fresh starter in a software engineering career. Those people, the work by them actually had some influence in my early software engineering work.

Yost: At the University of Illinois-Chicago, you were appointed the Director of the International Center for Software Engineering. Was that right after joining the faculty or later?

Chang: After joining the UI Chicago, I was thinking well, I did program analysis and I know how programs work. I know how to do incremental program analysis and incremental compilation; maybe I should get away from that at least for a while, and start looking at another dimension, which was the specification of software modeling. So I decided to go away from that, but then I came across some papers in software testing, that software testing actually can be supported not just by program analysis, but also by analyzing the specifications. So from specifications, then try some test cases, and usually those cases tested poorly. That's how I decided it would be a good new direction for me. So I started; so I went into software testing and did some papers in software testing. I ran into a gentleman from Fujitsu in Japan, at a conference, actually at COMPSAC — I will get into COMPSAC later, a little bit more on that — at this conference; the conference at this time always had lots of industry people coming together with academic people. So I arrange with Mr. Kiyoh Nakamura; now Nakamura-san and I talked and I said I'm doing this interesting software testing based on specification and I think it's a very useful research. He said he could support my research so this was probably the first instance for Fujitsu to support a foreign institution to conduct research for them. So I got some pretty big grants at the time; it's a quarter of a million dollars to start. So I started a software center. I called it the International Center for Software Engineering and then my focus became software testing and I used Petri Net as the specification model, and I think I published the first paper in Petri Net slicing, which I think is some of the early days, people began to understand that Petri Net can be very useful as technique for software modeling. It happens that at UI Chicago, we have another full professor, Tad Murata; Murata is well known in Petri Net theory; in the Petri Net area. I think he was the recipient of the first Petri Award, the Dr. Carl Petri Award. So I a learned a lot from Murata; he showed me all his personal literature; and then, of course, in return I showed him how we can apply Petri Net — it's not just a theory — to software development. And so I used Petri Net as a specification and modeling technique, and also I was trying test cases from specification, which is specification-based. I think those years we had very good cooperation and he supported my students, as well. And then I also introduced Petri Net, also in general, the specification-based testing to industry such at Fujitsu.

Yost: Did all the funding for the center in the early years come from Fujitsu or did you also seek additional support, support from federal agencies?

Chang: Mostly from Fujitsu, and then later, I also had some other companies; one in France, some in the States; also because of the vicinity, I would visit the Bell Lab very often. I also spent a summer there to help introduce Petri Net to Bell Lab people, I guess, I received a grant or donation from Bell. So I think it really involved still continue engaging myself with industry people although I moved back to academia.

Yost: At what point in either your career as a student or entering the profession did you join the Computer Society?

Chang: Okay. The Computer Society as a volunteer, I started in 1979, at the time I joined Northwestern University as a student. I talked about my professor, my advisor, Stephen S. Yau. He was actually was the 1974-75 president of the Computer Society; way back; in the early, early days. Then after he stepped down, after he finished his presidency, at that time he began to look more into software even though he was an EE type of scholar before. Then he saw that software was becoming more important so he started this conference called the COMPSAC, Computer Software and Applications Conference; the IEEE COMPSAC. He talked to Dick Simmons, who was the president after him, and they said we should start a new software conference, so they launched COMPSAC in 1977. And then in 1979, I went to Northwestern University and then in September, Professor Yau said I need some student helpers for this conference. So I said oh, I volunteer, I can help. So that's how I started serving the Computer Society as a volunteer; so we started in 1979, I graduated in 1982. After that, I began to learn from Professor Yau to be active, as he was very active in the CS. But every time he had some opportunity and asked me if I could help. I would say yes, I can volunteer. So I learned how to organize panels, organize sessions, review papers, all this conference business from him. So he was influential in terms of my career as a volunteer.

Yost: Can you describe that first conference in the late 1970s, and how it evolved in the early 1980s; how large was it?

Chang: Yes. COMPSAC is a very unique conference. I would call it a flagship conference for the Computer Society since the first year, since day one, it's a Society conference. COMPSAC has never been under any TC; COMPSAC is always a Society level conference. In the first 10 years, COMPSAC has — I don't remember the exact number — probably in the range of a thousand people. A lot from AT&T because even while I worked at AT&T in 1982-84, when COMPSAC got help from the company with internal routing, I asked them to sign up for COMPSAC, in those days. So COMPSAC was always in Chicago the first 10 years to allow AT&T let people to attend the conference, besides the academic people. So going forward, it's not just software engineering. There's a misconception about COMPSAC. Sometimes people classify COMPSAC as a software engineering conference, it's a wrong classification. Even today, some people don't realize that COMPSAC is much broader than software engineering. It's about software technology and in the past few years it's about computing at the core of any computer system. It's about computing. Of course, computing nowadays has lots of software, very intensive use of software. So that's COMPSAC. After 10 years, COMPSAC started moving around the world because after AT&T's divestiture and the company started breaking apart, we no longer could attract that many people from AT&T's Bell Labs, the economy was not that good, so we decided to diversify the audience, the base, and started moving around the world. Professor Yau was running the conference all the way through to 2005. COMPSAC 2005 was held in Edinburgh, U.K., and actually, in 2003 Stephen Yau started asking me whether or not to take it over; he was just getting tired of it. Running a conference for many years, certainly, there's lots of effort. So he asked me to run it. I said no sir, I'm too busy; I was elected as Computer Society President at the time, so after 2004 I finished my year as president and in 2005, he asked again and I had no other excuse. So I said okay, I'll take it over. Then I took over COMPSAC in 2006. COMPSAC has a two-layer governance structure. First layer is called the standing committee, so I put myself into the standing committee as the chair; made sure it happened every year; and long range planning. And then there's also a

steering committee, which is in charge of operations for a particular year. So the first few years I served as chair of the standing committee and the steering committee. I tried to transform COMPSAC into a more broad-based and more international flavored conference. Until this year, finally I finished the Kyoto Conference and I decided the next year that the steering committee needed to be someone else. That will be in Sweden next year, so I have Bruce McMillin from Missouri Science and Technology to be in charge of the conference so I can gradually phase myself out.

Yost: Was that successful, to broaden it internationally?

Chang: Yes. COMPSAC in recent years has really broadened its scope of support its base. We have lots of volunteers from Europe, from Japan. The program committee has about 120 members. These are active members in the sense that we do ask everyone to review papers. For each paper we always require three reviews; three quality reviews. If the review is not of high quality we send it away for another review until we have three. So it's a very high quality conference, very rigorous review, we still have face to face committee meetings, which is not easy to do these days because you end up traveling to a conference twice, attending the committee meeting, and attending the conference. But we noticed in recent years, we always have like maybe 40 people come to the face to face planning meeting. They spend two days selecting papers, design the panels, all these things. I think it's a very, very dedicated group and very high quality people. The review is always the strength of this conference, and then we also have many workshops, these are loosely coupled with COMPSAC but they come back every year, so co-located, loosely coupled or loosely affiliated workshops. This year was a huge year in Kyoto in July. We're having like 21 or 22 workshops, so it is pretty interesting, a very dedicated group, very good community. It's a well run-run conference, I would say. People just enjoyed this community. Actually, I've forgotten, too, that the conference has been co-located with another conference since 2008. It's called SAINT (Symposium on Applications and the Internet) jointly sponsored by IPSJ (stands for Information Processing Society of Japan) and IEEE. So it's a conference co-sponsored by IEEE; actually, the Computer Society, and our sister society in Japan. Since 2008 we co-located

these two conferences until 2012. In 2012, the organizer for SAINT and organizer of COMPSAC sat down, getting closer to the co-location, and the topics are very similar, nowadays. SAINT stands for Symposium on Applications on the Internet; and nowadays computer software is largely internet based applications, so you can see these committees are getting more and more closely related. So we decided maybe we would just merge these two conferences. So then this year it was merged and we have people from the SAINT community and the COMPSAC, still under the name of COMPSAC, so it's become a larger base. We look forward to further growth of COMPSAC in future years.

Yost: Has it remained a high percentage of people from industry?

Chang: Yes.

Yost: So the mix of academia to industry has remained pretty constant over the years?

Chang: Yes, if you look at the people serving on our committee; people taking leadership roles, the so-called chair positions, about half of them are from industry. We still are very attractive to people in industrial research. We are not really a practitioner based conference, or a trade show type of big crowd conference. This is actually very technical, very I would say, it has its technical depths. And so industry people, industrial research people are always part of the entire operation. So I would say every year is about 50/50 in terms of attendance. And also at COMPSAC, we/I begin to involve several presidents since 2009 into COMPSAC operations. For example, this year, Sorel Reisman, he's the 2011 Computer Society President, he's the general chair this year; and John Walz is our past president this year. John Walz will be the 2014 general chair. So getting presidents involved into COMPSAC operations has been helpful, so they understand what COMPSAC is about. And it forms a real friendship, *per se*, flagship conference for Computer Society. And we started also broadening our horizon by engaging other IEEE entities into future planning and development. For example, IEEE CCI, Cloud Computing Initiative, they co-sponsored the reception this year, under Steve Diamond, so there is more collaboration with other initiatives in the society. So COMPSAC will join also with

special interest groups and special initiatives. That's how we are going to plan for the future COMPSAC.

Yost: Were you a member of the editorial board of IEEE *Software* before becoming EIC?

Chang: Yes.

Yost: When did you first become involved with IEEE *Software*?

Chang: I got an invitation from Ted Lewis, who was the Editor-In-Chief at the time. I don't know where he got my name but he sent me an invitation in 1988. Oh, I now remember; I did a very, very good review for him, as a reviewer in 1987. He invited me to serve on the editorial review board, I say oh, sounds like a very interesting position. So I went to his editorial board meeting in 1988 and later he asked me if I would consider being the trainee for the Editor-In-Chief position. So I was appointed Associate Editor-In-Chief; I was still Assistant Professor at the time. So that's how; I really appreciated Ted giving me that opportunity at the time. In 1990, it was an opportunity because he would have finished his four-year editorship so he asked me if I was willing to be considered officially as a candidate for the EIC. So I think I wrote a very good position statement, about 10 pages.

Yost: Can you tell me about that?

Chang: Yes. Actually, for Computer Society service, I view every opportunity as a precious opportunity for me to grow, to learn, so I really take it to heart. For every opportunity I like to do my best work so I studied the articles published in *Software* carefully and examined the composition of the editorial board, the way the editorial was developed and I just started doing a lot of analysis at the time, how a *Software* issue is a bimonthly publication, how they crank out each issue. So I think I wrote a very thorough, careful analysis, also my vision; how if I was elected as EIC how I'm going to grow

Software. So I remember I wrote into my position statement about how can we actually involve more industry participation and contributions. That's because my industry background I always thought this is very necessary, especially for *Software* as a magazine, it should really serve the entire software industry, not just academic people. It should not just take the submissions by academia, right? So I wrote into my position statement that I would actually approach industry and write a report. That's a key part of, I would say, success of my next four years as the EIC.

Yost: You said you evaluated the strengths and weaknesses. Was the lack of industry contributions and participation one of the major weaknesses you saw?

Chang: Right. The weakness I saw was that we don't need to just wait there for submission. We should really go out, proactively go out and develop or cultivate the authorship, and not just academia. We should cover what industry does. This is something we should do. And so the efficient way there for submission; most of these submissions will be from academia. That, then the software industry still has a distinct industry flavor, so in order to really serve the software industry you need to know what are the real practical issues there, and what are challenges, and then think about people, the industrial people and so on and so forth.

Yost: So you brought a more aggressive recruitment, especially from industry.

Chang: Yes. After I was appointed an EIC the first thing, as I said in my position statement, I would compose the industry advisory board, and that's the first one in the whole society; in the whole IEEE there's no software advisory board for Society at the time. So I started my first few months, I did very, very active recruitment. We got a very, very good response; very good people like Bob Martin, he was the Senior VP for Bellcore. He was in charge of software systems development; I think that's how it's called. We had Dave Decker, he was the director of GTE Labs. We always had some middle management folks from all over the world; from France, from the Netherlands, from Japan; so it was a global composition of people who are concerned with software

development, the software development industry. So after I almost completed my final term I counted, I have about 30 industry representatives serving on the board; and perhaps we have about 20 editors, so it's a huge base, a very powerful group. So then, and this is funny, in 1990, the economy was not good and the Computer Society had budget problems. So I, as a new EIC, I knew nothing about the operation, how they make a budget. I made a proposal to the VP for Publications Ron Hoelzeman that the EICs report to him. So I said, may I have some money so I can have a joint editorial board and industry board meeting. He said okay, tell me how much you need. So I made a budget and submitted to him and then said I would like my first joint meeting to be held in Maui, Hawaii. That did not go well. [Laughs.] But Ron in the end was very supportive. He OK'ed it for me; he said why is this necessary; I explained and I got the money I needed. I got I think about 10k in order to hold this first ever joint IAB-EB meeting in Maui, Hawaii. Lots of people still remember that the first encounter between editors and — of course, there were some editors also from Bell Lab, and from industry — but we had a very dedicated group of people coming from industry, they just want to help *Software* magazine, you know, to help direct or guide its editorial development. So after a very, very cool meeting and actually, those four years, the industry played a very, very big role in editorial development. We had no shortage of topics, emerging issues in the industry, and they volunteered their own staff, their own technical people to help edit special issues. You could see all this swing this way, that way. So I enjoyed those four years as EIC. That model, actually, later propagated to other publications in the Computer Society.

Yost: What special issues are you especially proud of during your tenure as editor?

Chang: Many issues; one with CMM, Capability Maturity Models was pretty hot at the time. Nothing's perfect, you know, that allowed some important discussion. That's many issues involved in industry participation, direct participation. CMN is one thing; something on project management, of course — see that plaque over there — my work with *Software*. But soon this interaction with the industry people really would help the editorial board defined what is the real issue they are facing and we try to find what are

the best practices to deal with problems and what are the academic techniques that can be used for those things.

Yost: In 1994, you wrote an EIC column in the magazine called "The Changing Face of Software Engineering." Can you briefly explain what you put forth in that column and the reception to it?

Chang: Yes. I wrote several EIC messages; that is one of them; 1994. So I asked myself, as a professor, and I still did very active research in software engineering. And then it just happened that, to me, it seems for some people, some folks, in academia, they don't really appreciate software engineering. They think software engineering research is like other engineering, it is dry, it is rigid, but I began to see the possibility of embracing the results in other fields of study, you know, such as the soft computing. So I started; then I decided to write that special message. I see, for example, soft computing and software engineering can become more closely related, software engineering people can use some of these ideas from soft computing. Researchers ask and solve engineering problems. Myself, in particular, I look into a genetic algorithm approach; so evolution computation, fuzzy logic, a newer math, this is all the soft computing domain. And also those techniques had already been around for some years and there were some very good results and why can't software engineering researchers use those results? That's how I decided to write that article. I made some projections and it happens that 20 years later I look back and those projections all become true.

Yost: Can you elaborate on what the projections were?

Chang: For one thing, the genetic algorithm or evolution computation and the software engineering; I came across genetic algorithm research in 1992, I believe, and I think I published if not the first, one of the first few papers in applying genetic algorithms as an optimization technique to solve software engineering management issues, in particular, task assignment. You know in the software industry, every day when managers go into the office the first thing is to decide who should do what and who has done what, those

are management issues. So, "who should do what" is a task assignment. But who has the expertise to do what? And who should be phased in to do what? These all are optimization issues. Software engineering research at the time has not really benefitted from the genetic algorithm or any of the evolutionary computation techniques, which can be very useful for optimization of these kinds of task assignment issues. So that area, actually, began to really take off in 2000; maybe 2001, 2002 although, I looked into the problem in 1992. I think eventually this kind of nature-inspired, or I call it biological reflection — is the term used in my article — is going to take place, is going to grow big. And then in recent few years, there's a community called Search-Based Software Engineering, SBSE. You look at their website, they have already collected more than a thousand papers using EC, evolutionary computation, in software engineering, especially search-based optimization. So that's one example of my projection of — I think this area is going to have some merit — of course, it become true, as I predicted in 1994.

Yost: You mentioned, too, that in addition to that message in your 1994 EIC column, you had several others. Can you tell me what those were?

Chang: Yes. That's one thing. I need to go back; this one that just happened to be one that was also in my research area. I mentioned a few things besides this biological reflection, using the soft computing. I also believe — you need to give me a moment. [Laughs.] You'll be able to edit this?

Yost: Yes.

Chang: A few things that I; It's not open source, but I don't believe the software crisis will ever occur, that's just because you have more capacity, they start to expect more. Then you have more capacity, they start to expect more. So software crisis is not happening, in my biased opinion.

Yost: So all the press over the years about [pause]

Chang: Yes. People just want to make a point and make a living. Yes. If you can stop for a while, I can go over that, if you have interest in that.

[BREAK]

Yost: What did you find to be the greatest challenge or challenges in running IEEE *Software*?

Chang: Running IEEE *Software* is actually, in the beginning, is a big challenge. I think our recruiting, Ted [Lewis], decided he's not going to accept any more papers so I can have a new queue, so I have my own clean slate to start with my editorship. So we only have about three or four papers left in the queue when I started. That's very scary so the managing editor, Angela Burgess, at the time, we worked together in all these years. Angela didn't get scared, but she asked me; she emphasized we have to generate; after 1990-91 we don't have any more papers so I need to really start to take immediate action to start recruiting papers. So I went to academia, I went to Bell Lab, I went to lots of places to solicit where I know people have good stuff. So we have to just survive the first few issues until finally we get the industrial board people to come out. The problem with the industry board is that sometimes they have a good problem, they have a good contact, but still the editorial decisions need to be made by the editorial board not an industry advisory board, because they are advisory. Although they are very, very helpful; very supportive, and they contribute lots of resources, I need to really know how to balance between the two. Fortunately, people understand, but we had to get people to understand, the industry people understand they are always advisory. That's one particular challenge. Besides, there is often; in my term, actually, the problem is really not short of papers but rather how do you mix and match each issue? If I have a special theme, how do you get the best perspective on the issue? So it cannot be too academic, cannot be too industrial/practitioner based, there has to be independence in the point of view. In those four years, our slogan, or tagline, is *Software* is where the industry needed academia, of course, and a new tagline is, I think, *Software* magazine is where the doers think and

thinkers do. And that is how we struck a balance between two different camps. And we faced the challenge and I think we actually served our purpose pretty well.

Yost: That's a very useful slogan.

Chang: Right.

Yost: In 2003 you became President Elect of the Computer Society. Can you talk about different roles with the Computer Society you had after IEEE *Software* but before becoming president elect?

Chang: Yes. During that period before I was put on the ballot for election for the president, I finished my term as the EIC for *Software*. Then I was included in the election for the Board of Governors. I ran for election twice, so six years serving on the Board of Governors. In those six years I began to get involved more, I got a very good view of how the society actually operates, besides publication. They had many other activity boards in the society. So the first two years I was put into the audit committee; you know, the newcomer, a fresh face, you just started with audit. And after two years, then I asked, can I be moved out of audit, I'm ready to do something else. Actually, I think at the time, Society leadership — Barry Johnson, Doris Carver — in particular, Doris asked me to serve as the secretary of the Board of Governors. So I had a chance to sit right beside her — she was the 1998 President of the Society — then I observed and learned how Doris ran the society and I could really witness the flow of everything by sitting right next to the president. So I am really thankful to Doris for giving me that opportunity. And then Leonard Tripp, President Tripp also was very kind, he asked me to serve as the VP for Press Activities. That's the first VP job I got. Of course now to me, that's a new challenge, a new direction. Of course I was very pleased and the only thing is that by serving that year as the VP for Press is to take the charter to eliminate myself, in the sense that Press Activity was not doing well, in terms of profit and loss. Actually, it's been losing money for years. So the society decided that press activity needed to be eliminated and move press into the publications board. So this, actually, I'm just joking

that I eliminated myself, but actually, that particular assignment gave me the opportunity to see how the things worked and I got to see how to make a business decision in the society. So that actually gave me a good opportunity to really learn about the society, not just as a volunteer organization, but also as a business. You cannot just keep running into deficits and still be able to serve your members. So I learned that point. So I think Leonard gave me this opportunity; and I always appreciated it; I was always joking that I eliminated myself, but this was opportunity to actually grow from that assignment a lot. So Doris and then Leonard Tripp, they both had very, very big influences and gave me the opportunity to grow in the society. After that, I merged press activities into publications. Then I started wondering what should I do, the next step. And then the society gave me the opportunity to become the VP for Educational Activities. That's before; and then finally, I moved myself onto the ballot for the president. So that was in 2002 and 2003; I served for two years as the VP for Educational Activities. In 2001 I actually ran election for the First VP position for Educational Activities. No, I think I ran the election for the First VP and then after I won the election, I chose to stay as the VP for Educational Activities for another year.

Yost: What were some of the critical issues or initiatives for education?

Chang: Education is interesting. Again, that helped move me into yet another area besides conferences, and press, and now education. Of course, as an educator in academia, that's kind of like my home. So I met with Willis King and all these very nice folks, and then one big project at the time is computing curriculum. It's called Computing Curriculum 2001. I actually started this project before I became VP for Publications because I think we started in 1998. When we first started, Ben Wah, he was the 2001 president, he asked me if I can take the lead in this project. So we went to talk to ACM folks and said, now, since curriculum 1991, is almost 10 years old, can we not join forces and start a new curriculum effort, so we started the new task force. And then after two years of effort, lots of meetings, people follow the work, we came out with a CC 2001 report, and I removed myself from the chair of the task force, because I was the VP as I had to approve the report. I cannot also serve as leader of the report and approve my own

report. That's why my name was moved into the VP for Education and then after Jerry Engel, I believe, he came up to serve as the chair of the task force, representing Computer Society. So that's one big project, I spent a lot of time on that. And the effort put forth by volunteers, precious time, and lot's of resources for both societies was considerable. That is, I think, very influential in the world of computing education and later we keep on; but that report is more of a science report for computer science curriculum. And later they carried on and completed other reports, several volumes; the Computer Engineer Report; the Information System Report, that was actually done by another society but we adopted it.

Yost: Were there any differences between the Computer Society and ACM culture that manifested in the interaction in working on the curriculum?

Chang: Actually this particular effort started way back I believe in 1968 that the two societies, and that is Curriculum 63; Curriculum 1968, 1973 it was also the committee [pause]

Yost: A very long history.

Chang: Yes, a very long history. Every so often they got together and this is really the entire community business. So in 1998 and 1999, I went and talked to ACM folks. I went and talked to Richard LeBlanc, he was the VP for Education for ACM, and Rich LeBlanc said okay, good. This project is due so we can work with Russ Shackelford at Georgia Tech, and then we also involved Eric Roberts of Stanford, this represented the ACM side. And then we had our folks; Jerry Engel, Willis King, also Pradip Srimani, and so we formed a task force. It's a community-wide thing; we involved lots of people not just from major universities but also some other liberal arts colleges, because curriculum has to be good for all kinds of institutions, it's global, world-based.

Yost: When you decided to run for president of the Computer Society, what did you see as the most important areas that needed change within the Computer Society?

Chang: That's a good question. I think every president that comes on board has his or her own mission and agenda. In my year, after I was elected, the first year you are called the President Elect. As the President Elect you have the opportunity to work on bylaws of PPM, and of course, Computer Society president is a three-year term. Most people think it's just one year, actually it's not. It's a three-year term because the first year you serve as the President Elect and you have about one-third of the society business to take care of. So I spent my first year looking at PPM, and also looking into how society is doing, of all. And my vision at the time was this society has long been a membership society in the sense that the Society is about the community. Community is about people becoming members and then we have lots of activities for the members. I began to see that there is a need to transform from a member-based structure or society into a service-based organization. So I had the opportunity to lead the effort, as President Elect, through my year, to conduct another round of strategic planning called SP-5. This is [pause]

Yost: A five-year plan?

Chang: No, it's actually every three years

Yost: Oh, so just the consecutive numbering.

Chang: At the time SP-4 was before I became the President Elect. So SP-5 task was right in front of me so I started learning how to do strategic planning for a not so trivial organization. The Computer Society at the time would have an operating budget of about $31 million, which is not an overly large budget, yet, there is 10,000 volunteers working there for nothing. So it's actually a very big and complex organization. I need to lead this effort, this task force, through strategic planning. So I finished my job with SP-5 because that's the fifth iteration. SP-5 turned out to be pretty robust, because after SP-5 it took us about another five years to do the next one, because SP-5 calls for transformation from the membership organization to a service organization. Why people would like to become a member? Because you have good service. So someone says you have good service then

people will be drawn into this community to become a member. So SP-5, we actually route SP-5 to the entire organization after we finish it. I also brought SP-5 to China, our sister society in China called CCF; Chinese Computing Federation, or China Computer Federation, CCF; and show them how you can actually grow; China is a very different operation. There are no individual members at the time, but they just start getting the freedom to be able to start to enroll individual members at the time, but how to do it? So I shared my SP-5 with Dr. Du. Dr. Du was the Secretary General of CCF and I showed him this SP-5, actually made it be a service-based organization, so that people will feel that that in return for a membership fee, they receive services provided by organization. So that was my President Elect year.

Yost: What are some of the most important components in transforming the CS from a membership-based organization to a service-based organization?

Chang: Two things. We really need to start reorganize the Computer Society so that we would have many service bricks — that's a term I use — but later I found out that's actually like a web service nowadays. You have services, then you compose services; so you will break it down into like atomic services; so atomic services is what I mean by service bricks. So you need to have those fundamental units to provide service, and you don't need to get aware of that organizational structure. That's one thing and the other thing I call one stop shopping. As a volunteer, sometimes it's very painful trying to work through the organizationl structure, which should be a black box, but unfortunately, the way we organize ourselves is a white box. So the volunteer has to remember who to contact for what task, even running a conference. Even as of today, to run COMPSAC, I need to remember who is in charge of credit card, CB card, in the IEEE organization; who is in charge of our CB bank account. And then I need to know who is the MCM manager this year, who is going to make a report this year, who is the other person doing the VAT business. When the conference goes to Europe in particular, you know, you need to take care of the VAT issues. So all this ends; I still have to know who to call; and then when I was traveling in Kyoto, I just cannot remember who is the person I have to call upon to help solve my credit card issue. You know I cannot get all of these banquet

events without paying and the card did not go through. So sometimes the staff is very polite and very helpful, but it's an organization issue. So if we can help our volunteers or our members; it's a one-stop-shopping concept. You just visit your contact and this contact, this person, will go through the maze of the organization and help you find out the right place or person. So that's another concept. I think Angela Burgess picked it up; I think she has really these past few years, transformed the society [to be] very, very service oriented, and try to give us; implement this kind of concept so we can just contact one person, talk to the person, if you remember the name, and that person will help you navigate through the whole operational units in the Society.

Yost: Did the strategic planning that you led, set a real kind of framework for what you wanted to do in your year as president and then past president?

Chang: Right. Yes. I wrote my roadmap; I just tried to implement according to the plan. So SP-5 became the blueprint for the transformation process of the society. Then through my year, I just started propagating the concept. Of course, SP-5 has to be approved by the board and the people need to understand what I'm talking about. I see it — not just during my term — but later presidents, during their year, they also start engaging office services. We have many office services provided to serve the members better. Of course, actually during Ben Wah's term, he already started signing agreements with the publisher for online books. Later, we also have all kinds of training programs, certificates; these began to take place in the society as a critical service for the computing professionals and software professionals. In particular, I think I began to look more into CSDA [Certified Software Development Associate], CSDP [Certified Software Development Professional], during my president year or the past president year. I traveled to China, for example, try to sell them the certificate concept with limited success. China is still not very appreciative or open; maybe nowadays they will begin to see international organization become a more important topic. But I tried; believe it is the core service. For example, CSDA, CSDP is a software engineering certificate and that is also maybe a way to attract more members from China — it's such a big country — but it is so we can have better cooperation with CCF, so that they can sell the service on our behalf. So I can see what I

said in SP-5 is we need to become a service-centric organization. I think it's now taking shape.

Yost: Was there any resistance within the Board of Governors to the vision of SP-5?

Chang: No, I didn't see any resistance. People began to understand, yes, you cannot just rely on a membership fee because it only accounts for, I don't know, only eight percent of the entire budget. We do a lot of other things. It's actually more and more service; we just don't have the organizational structure to support the service-centric mission. So I didn't see much resistance, people understood it. But how to do it? And how long it takes for the transformation process. I was just president for that one year. So it takes so many stages, so many generations to get it done.

Yost: Conferences, obviously, are a big part of service and an important source of revenue.

Chang: Right.

Yost: What about technical standards? Is that something that generates a lot of revenue for the society?

Chang: Standards, that's my only weakness [laughs], in terms of running the Computer Society as an organization. I never involved myself in SAB, the Standards Activities Board. Of course, I heard things. I know they actually generate lots of revenues. Unfortunately, the revenues are generated for the IEEE central organization, the Computer Society, I don't think they have a share of that revenue. That maybe still true today, but don't pen my words, I have been out of this budget process for so long. But yes, I believe standards is a very important part of the Computer Society volunteer work. It generated lots of revenues, but where the revenue goes, during my term I know it all went to central IEEE.

Yost: In the second half of the 1990s, as you became more involved with different parts of the Computer Society and started to see the big picture of the organization, what was its relationship with the IEEE? And did that change between the mid-1990s and the following decade, in the time when you were incoming president, president, and past president?

Chang: I think it's fair to say that the IEEE and the IEEE Computer Society, they grew together during about the same period of time until maybe the year 2000. I think there's lots of discussion at the time how the synergy can be built between IEEE central, IEEE and the Computer Society. The Computer Society has its own operation, its own staff structure, its own conferences, own publications, and it's not small. At the time, it's about one-third of the entire IEEE. IEEE has many other constituencies, many other societies, and then, of course, the Computer Society is still part of the TAB, the Technical Activities Boards under IEEE. But the Computer Society is such a critical component of IEEE so I think there was some struggle in the beginning when we started this social agenda; between two camps, two leaderships. I think for several years there were lots of discussions, meetings, debates, and struggling. Then I began to feel that people start taking a more open-minded approach and things began to get solved because after my term, the coming up, the new leadership after my term begin to take a more proactive approach. Angela Burgess has been and is a big help; she really helps communicate between two ends. I think up to now, I believe that's very good; much more cooperation between the Computer Society, as a society, as operational unit under the IEEE umbrella and IEEE.

Yost: Digital services and digital delivery of content become increasingly important. Can you talk about changes that were underway during your time in leadership?

Chang: During my time, we started a new, it's called electronic publishing. I think that's what it's called. People began to realize that that's the trend and we have to start moving into, on the publication side, open access. So it takes 10 years to finally get the problem under control but I would say, after I stepped down, I mean I finished my term as

president, and during my term as the EIC *Computer*, that's when the transition started taking place. In *Computer*, I remember in the editorial meeting, we have always on our agenda how do we publish computer articles online and how do we collect the metrics, and access, all those things. But after finishing my term I believe we also have *Computing Now*, as the portal for not just *Computer* but many other important technical contributions all collected under one umbrella now. I think the society has been pretty successful in moving into the digital media area. So I think the trend is good and essentially we have that under control; although I'm old fashioned. I have to admit that when I read a paper I have to print it [laughs].

Yost: I do too. Academic libraries have been a powerful force as well as legislators on behalf of tax payers. There's been substantial talk within Congress about the importance of open access of publications funded by federal projects.

Chang: Right.

Yost: All of this, of course, is well intentioned, but it also poses a threat to a revenue stream of the Computer Society, doesn't it?

Chang: Right, it does. Speaking for myself, I'm all for open access. I believe that information, after all, has to be useful for people. It has to be open to people who are doing research. On the other hand, how do you actually support that concept? And the Society, the IEEE as a whole, had been struggling with the issue. I think now they have a kind of hybrid model, so some sections of publication you can have open access mode, and some has a hybrid mode, some has still the old-fashioned mode. I think we're still trying to find; to strike a balance. How do you; somehow you can have enough revenue streams to sustain your operations for members. On the other hand, the trend seems to be that people want to be able to access material. It's not just a budget issue, a financial issue; it's also the quality of the publication, the quality of the articles. If it's an open access, is that going to have a poorer quality, are we going to sacrifice quality? And asking the authors to pay a thousand bucks an article; is that really sustainable? Myself, I

think although it's a trend we are moving to, but there's still lots of details that need to be ironed out. That's why actually we still have this hybrid model as the pilot program.

Yost: During your time as president, were there any major changes with conferences and how that portion of the Computer Society operated or any pivotal new conferences that were initiated?

Chang: During my term, actually, I did not do that much with conferences, in terms of changes to operation. Of course, I've been involved in conferences since 1979. After I finished my term as president, in 2006 I began my work to lead COMPSAC year after year. I began to take a much closer look into TMRF budget model, a whole different kind of operation. I do want COMPSAC organizers to have one-stop-shopping option but I think I have been very critical about how our staff supports the conference organizers, not just through the Computer Society staff but also through IEEE staff because I ran through every step of running a conference since 2006 so I encountered the problems, I noticed the problems, I always reflect as a past president of the society. I think I need to tell them what I really [pause]

Yost: It was a bit confusing and bureaucratic to work through?

Chang: Not necessarily bureaucratic; the staff are nice, okay? It's really how to reorganize ourself so that it becomes really very clear and we always are more supportive to the volunteer from the organizational viewpoint. But when you come to staff, they are always professional, they are very nice. But I ask them to; for example, as a conference organizer, we, of course, are concerned with our final budget status so I wouldn't create a big surplus; I don't want ever to go into deficit. But the Society due process — TMRF — is that the Society only charges 20 percent for overhead. So that 20 percent over the entire expense is actually a lot of money and I happened to be able to handle that most of the time. Only one year, after paying the overhead, I kind of went into a small deficit, so to speak. It's really not a deficit because I paid a large sum of administrative overhead. But as a conference, you pay 20 percent overhead, in return you need to get service that

you're supposed to get. If volunteers don't feel they get the service they are not willing to pay the 20 percent overhead. So I have to raise those kinds of concerns and issues. One funny thing about the old TMRF, just as an example., if we have bad debt — bad debt, in the case of COMPSAC, means the IEEE service, MCM, Meeting and Conference Services, they fail to collect the registration fee or some revenue, and that was reported as a deficit — it's a debt, not deficit — into the debt cell, and that becomes your operating expense. Then as a COMPSAC organizer you have to pay 20 percent overhead over that cell, so you see what I mean? So that is the failure of collecting revenue because actually IEEE service did not collect it; so when it comes to organizer, how come I have to pay 20 percent penalty because of error not committed by you.

Yost: Overhead charge for some revenue not received?

Chang: Yes. Those kind of things are just funny. That probably has not been solved. But on the other hand, I think the Society began to see a little more lightweight budgeting process for a more polished conference; So that's now engaging those lightweight processing; I am happy to see that. I think everyone goes through the heavy duty TMRF, you know this whole number games because that's the heck of it; you just fill in those numbers and then you pray for magic and, surprise, it's not a very good exercise year after year for COMPSAC organizers.

Yost: Can you talk a little bit about professional certification programs and to what extent that was an important activity and issue in your time in leadership?

Chang: Yes, in particular for software engineering. I think this should go back to when the SWEBOK started, Software Engineering Body of Knowledge. Again, that is under the leadership of President Leonard Tripp. I was invited to participate in that effort. Actually, I served on the Industry Advisory Board as a professor from academia. But anyway, then he put a confidence in me and I contributed to that discussion. Then eventually, they came up with a guide to the Software Engineering Body of Knowledge. Since we have the body of knowledge published for software engineering, naturally the

next step is how this can be useful or influential in software professional work. So they started looking into a certificate program for software engineers. This is way back, the year 2000 or earlier when we had this concept. We moved this proposal to IEEE; unfortunately, the people in IEEE, as engineers, some people don't necessarily recognize software engineering as engineering discipline. That's why we cannot call it Software .Engineering Certificate; so it has a weird name; it's called Certificate for Software Development Professionals. But I think that's a very important and useful program, not just for the U.S. software professionals in this country, but also I think for the whole world. Software professionals, at large, need to have some certificate, some kind of evidence they can develop software with good quality, and so on and so forth. That's why I believe that the software engineers in China ought to be certified because more and more software is outsourced to China and we need to know the quality of those professionals. So I think now I believe that CSDP; and later they have another program called CSDA, because CSDP — it's not easy — CSDP exam, I don't know if you have ever taken a look, it's not an easy test. The people with years of software experience think through all the battles, struggles, problems, because the questions really ask very tough software problems, solve for difficult issues and problems. You need the experience to be able to answer those questions correctly. So they found CSDP is maybe too high a bar for many software professionals, especially from a different country. So they have another program, which is [pause]

Yost: Did it have a meaningful impact on certification and the development of the software and services industry in India?

Chang: India. I'm probably not the right person to ask about India, I know more about that in China. But I believe they did try to convince India software industry to also adopt CSDA and CSDP. Maybe more the CSDA, our associate program, just like a four-year degree versus a two-year degree, as an analogy. I believe it's working for both China and India. If they were to continue to be, not just as an outsourcing base, but also developing their own software industry to become more mature. This is a very useful vehicle.

Yost: You talked a little bit about your leadership with the Meetings and Services Committee, following your time as president. Are there other aspects of that that you haven't talked about that you'd like to?

Chang: Actually, I did not do much, I have to admit, after my one year short service as chair of the MSC, Meetings and Services Committee, which reports to the board directly, it's a direct reporting structure. But during my term, at the time, I think that IEEE as a whole began to realize they need to transform MSC into a different committee. I think later it was called the Conference Committee, so MSC no longer exists. Jerry Engel, if I remember, Jerry was 2005 President of the Computer Society. I believe he took it over and there were some changes inside MSC or actually Meeting Services, as a service unit. But actually, because of that experience, I began to understand how IEEE operates in terms of supporting meetings and services. So later, because I also kept heavily involved in running a conference, I gave lots of input. I think that IEEE made some changes in terms of how they operate, how they support, how they make financial reports to the different tiers. So my influence actually is based on that one short year of experience. I actually made some impact afterwards.

Yost: Can you talk about how you came to become the EIC of *Computer*; had you had any involvement with the magazine before?

Chang: Actually, I was not involved with *Computer* operation, not much. But, as I said, I always followed the footsteps of Doris Carver, and Doris took me to serve as her secretary, so I learned from her. And then after her year as president, later she became the EIC of *Computer* so I began to think maybe I can also help serve *Computer* as an effective publication of the Society. So I began to follow the editorial of *Computer* and how other people working for *Computer* — the editors, the staff — and also I talked to Doris. It seems to be a good next step to serve Computer Society, and Doris, she always stayed kind and supportive to me; and yes, she said, fine, come and observe. Eventually I put my name in as a candidate for the EIC in 2006, and then in 2007 I was appointed as EIC.

Yost: When you came in to lead *Computer*, what were the things you felt *Computer* was excelling at and what areas did you feel it needed to change, grow, or advance?

Chang: It just happened that I had experience as EIC for *Software* and I actually did something there in particular to engage the industry people to be more proactive in helping in editorial. So I had that mentality when I came on board and became the EIC of *Computer*. I believe, also in my position statement, I mentioned that I would like to expand on the industry connection in working with the *Computer* publication. And then, as I said, after I wrote something I meant what I wrote. So *Computer* is an effective publication, so there's no shortage of papers for publication from both industry and academia. But you look at it, publication mostly is still more from academia. I think that we need to strike a balance there. I think at that time, the managing editor — actually, the senior editor, or acquisition editor is Scott Hamilton, who passed away two, three years ago — he actually was very active. He'd go after best minds in the industry. He went to the NASA lab, he went to Google lab, he actually got some very good papers out of industry but it was through his personal effort, personal connections. But *Computer* magazine is a monthly magazine, you need lots of papers, lots of articles so I was thinking that maybe I should establish another industry report for computer. And I did that; I started connecting with company VPs, sending invitations, and later I began to realize that the caliber of the person who agreed to serve on this new industry review board, was good enough for the Computer Society. So I talked to the Society leadership and said can I put this new IAB into the board or maybe at least have them work for the Executive Committee? Executive Committee consists of all the VPs, President, and the ED. So they would come back with ideas and we have this new industry board I formed. You look at it the caliber of this board, this board is actually bigger than IEEE has ever gotten [laughs] because I look at IEEE, so-called industry report by IEEE, this is very different, very different category of people. So Computer Society embraced this IAB, then of course, my original intention is to ask them help with *Computer* editorial. So finally, the society, they decide to phase in IAB, another advisory entity to the executive committee, and then *Computer* will become the conduit for the industry, for members to

voice their concerns, the issues, and emerging topics, to reach out to the entire community of computing professionals. So I think that's what *Computer* is nowadays, that window for IAB to reach out to computing professionals. That is what has been accomplished, although it is different from what I originally anticipated, but its broad; they have not been heavily involved in directing articles, as I did for software. But this served a purpose far beyond the publication, *Computer*, it also helped other business of the Computer Society, like CSDP and CSDA, I think that is fine.

Yost: Can you compare and contrast *Computer* as the Computer Society's flagship journal and *Communications of the ACM*, as ACM's flagship journal?

Chang: Yes. *Communications* has transformed itself a lot in recent three, four years. I think before; during my term as EIC of *Computer, Computer* versus CACM, I think *CACM* at the time, was very, very information system based. Also, MIS. ACM, CACM, was really not a journal for computing at that time. *Computer* is really computing, computing technology, computing professionals. So I believe — what's his name of this editor for *CACM*? — Moshe Vardi. I believe he actually looked into how *Computer* influenced editorial and when he became the EIC, he transformed the *CACM*. He made it into a really strong competitor for *Computer*.

Yost: Do you think that's helped it become more relevant to industry?

Chang: I think as *CACM* become more relevant and the difficulty with *Computer* in recent few years, is that it has a very, very big reduction in terms of supporting staff, in terms of editorial staff. Of course, the Society has to become more efficient, so the publication office on the West coast, they began to use a different model. For *Computer* magazine when I was the EIC, it had six or seven dedicated staff. The society can no longer afford that so they had to change the way it operates. It has become very, very lean staffing, in the sense that you just cannot compete against, for example, *CACM*. I believe when the new EIC of *CACM* came on board, the ACM leadership had okayed $1 million to *Communications* ACM, in order to transform this magazine to become more relevant,

more computing-based. Sometimes *Computer* is suffering from the loss of budget and staff, so that's how we get to today's situation. On the other hand, the Computer Society has been investing a lot into *Computing Now*, which has become another good service to membership. I don't know if *Computer* is losing its reputation and prestige in the face of this new service tool called *Computing Now*, I hope it will not lose its esteem. On the other hand, it seems that the new digital media era; it seems we are in the situation trying to see where *Computer* should actually be positioned.

Yost: Was *Computing Now* in the idea stages—under discussion yet—when you were president or did that come later?

Chang: It actually came later; another president. Well, while I was the EIC, they already started a program called *Computing Now* out of this electronic initiative. But *Computing Now* became more mature after I stepped down as the EIC. So not very much overlap.

Yost: I'd like to return to discuss the evolution of your research. We haven't gone much beyond your early years in your career and your dissertation in discussion your research. Can you tell me about your software engineering research? I understand that early in your career you followed more of a classical approach with formal methods. Can you talk about that?

Chang: Right. After I finished my Ph.D. degree from Northwestern University, I began looking into modeling specification, to the level a little bit higher than programs. And I used to have to present models as a spec model, and in turn, had cases out of it. I did a model based on slicing. Slicing means you only slice out the most relevant part of the program or the spec. So that I did, probably, paper in modeling, in test case generation, 'til 1991-1992, I ran into this genetic algorithm paper. The first paper was by John Holland at Michigan, in the 1960s. I began to feel that well, this seems to be a very useful technique for software engineering problems, so I focused on task assignment, SPMNet. And then I believe that paper had some influence in the field and also people started using similar approaches for other things. In fact, the first few papers applying genetic

algorithms to software engineering were mostly in the software testing area because of test space, because of test case generation or the space can be very large so you need some kind of optimization technique. Also, people started, based on what I said about how genetic algorithm can be used for software management, I think there are some papers published in a similar direction. So that is perhaps through almost the year 2000, or maybe 1998. I also began to look into CSCW, computer supported cooperative work. So I started looking into; because the internet became more pervasive at the time; so I started looking into how can we cooperate better, more on the internet, for software development? So I looked into different kind of cooperation models, net-centric cooperation models for software development, agent-based software development model, because of the internet. I spent a few years looking into CS adaptive research, cooperation model based on the net theory. Again used Petri net for some of the modeling. I actually did a very nice project, I believe; it's very hands-on; it's called MeetingNet or M-Net. MeetingNet is the first generation of Net based Meeting. I actually also used a PBX switch, so that I can support the network, and Net based Meeting, and that became a very interesting project and students who graduated in that period of time, many of them went to Cisco, Bell labs, Motorola, as a result of that experience. And I went to AT&T about M-Net. I think you guys only do 800-based teleconference calls. This can be very useful because that's the future, but I think that's too far ahead. [Laughs.] On my own, I talk to them. They said no, and they asked will this generate a billion dollar business? I said probably not now; then AT&T had no interest. Nowadays with the World Wide Web, all this in twenty years, I notice we began to use such a teleconference, we also have what-you-see-what-I-see, all this network interface. So it took almost 20 years before people began to realize. So I spent some time looking to those kind of techniques in the cooperation models, and the tools, and then I began to — this is about 2000, 2002, 2003 — and then in 2002 I joined Iowa State University as chair for this [Computer Science] department. I served as chair here for eleven years — too long — so I decided to step down. So that's the research before I came to Ames.

Yost: You're one of the pioneers in search-based software engineering. Can you discuss that?

Chang: Yes. You know, software development is a very dynamic process. So before my study, I think people look at software, many have problems with task assignment, it is more from the static perspective. So you look at the programmers database — who has what expertise — you try to make match, and then who should do what, it's not that much dynamic factors were considered. Then software became more and more, more dynamic and we do need a different kind of technique so that we can always make it adjust to new situations, new environments, in more rapid manner, and so you need; and that means you constantly optimize new processes. So that's why I found genetic algorithm, a technique that can be useful. In my paper, for example, I mentioned in particular that people are not static. People work all the time. Software engineers evolve all the time. Your skill; you can gain skill, you don't necessarily stay at the office for 365 days; you may have vacation plan; you may need to take a maternity leave. So software organization, is a very complex organization, it is very dynamic. So in this kind of situation it makes optimizing necessary in order to deal with the dynamic nature of software development process. So I looked into these things. I have my model which will adjust to the new factors, evolving nature of the software process, software optimization, and you reoptimize every time and come [up] with a new assignment, new source. So that could be very handy for software managers.

Yost: In 2003, you branched into services computing research.

Chang: That's right.

Yost: What lead to this change in research direction and can you talk about the importance you place on the human dimension in understanding software development?

Chang: Yes. The effect of the software services is it's about software. With services you have more stringent requirements. For example, service level agreements, you need to make sure the service will fulfill what is promised, and even on the official agreement. So with software services and services computing, because of the internet and we start

talking about web services and SOA, Service Oriented Architecture. So I moved into the area to take a look. I was actually asked to — by Dr. L. J. Zhang, he started this service company conference — to chair the services computing conference. That is how I got involved into services computing, in general. And then I also helped build probably the first model curriculum in services computing, which is not proven yet, but we have the modules, anyway. And then I started thinking about how — this is, of course, you have a conference, you have community, and you have a curriculum staff — what should be right research to conduct. So I decided services needed to evolve, because again, this is serving industry and it's not a static industry, it's dynamic. That's why people work in web services, the most difficult part is sometimes to make a composition, how to dynamically compose web services. So I decided to look into what are the dynamic dimensions of service computing and software services. And that's how I start working toward this direction. I believe in 2007, I began to have this concept of support software evolution by optimizing the context and situations of a software environment. So I published that paper called 'Situ' in 2009; actually, the paper was done in 2008. I think that paper has started gaining some momentum and also people started paying attention to that paper because I basically said in the paper that look, service computing is about dealing with the context and the situations of your service environment and I have a formal model of — it's called — 'Situ.' And software service is meant to provide service per humans' intention to use the service. So in order to model situations, you also need to model how people feel they are situated. People are the subjects using the service. So I became more interested in the human dimension because of that kind of realization and I began to formulate my models; actually, in the past, when you are developing software systems, you are mostly developing the system dimension. If you talk to the user, or engage human, and you do repetitive study, you collect requirements from the user, after you have done that you still go back to your system domain, your system dimension. You feel more comfortable in creating code, creating system. Now you interrogate the human subject again; is this useful? You see the connection there? Because the software has to continue running and continue evolving, and that process, when I worked in Bell Labs, took two years. From this release to next release, and through all the usability study, field trials, and come back into your system dimension in the lab, and you do it; of course,

nowadays with a software company like e-Bay, Yahoo, and all, you have these patches every day. So this is evolution in a dynamic manner, dynamic fashion, and very rapid. How do you actually support that so that you can actually fulfill the human expectation for all requirements of using your service? It's not turning out to be very easy because first you need to establish what is human intention? Whether people feel satisfied with your service, the software service, actually on the fly or during their own time. How do you shorten this evolution cycle, in the service computing sense? That has become my major challenge in my research.

Yost: Can you discuss the impact of your research in this area, if colleagues have taken a similar approach? And/or has it been adopted within industry to any extent you can discern?

Chang: Actually, I'm not sure about U.S. industry, although I wrote a proposal along the same line to NSF. They did not fund it. I do not blame them because this is just a very, very ambitious direction. But then I went to China, in the past two years, and actually they paid attention to what I wrote. I ran into a software engineer after a lecture I gave he came to me and said that we, people working in this company — I cannot name it — we actually use; we looked at your paper. We actually apply some of the statistical analysis, statistical inference tools, and they had started to make sense out of it. I just have; a letter arrived two days ago, he said that we looked into your paper. We actually conducted a project, an experiment, we start making sense out of what you said and what can be done with a model. So my model is based on situation theory; so my work is influenced by people like Barwise, and also John McCarthy, Stanford people. They have this situation theory. I looked at those papers published in the early years, and also even the recent years. It's more mathematical than the logic-based theory. Those theories by a mathematician and a logician are not really computational, in a sense. So I needed a model; also very useful concepts and principles; I need a computational model. The way I define situation is not just a context plus some logical reasoning. My situation is defined to be DAE, a 3-tuple. "D" means desire, human desire. So we need to know what human desires to have, okay? They create a service to meet the desire. So desire is really a

human mental state; is a human expectation of using the service, which is the most difficult part, of course, because human desire is a hidden mental state. How do you make sense out of a hidden mental state? "A" is the action; and the "E" is the environmental context. So you have the behavior context, which is the action, the "A"; and the "E" is the environmental context, the thing that completes the picture like sensors and the data; and the "A" is the activity. You recognize what action the person, the human subject, or the activity engaged. So if you have at least such a 3-tuple, you can have a good theory; a trace of all these three through a window of observation; then for an end-to-end scenario, in the sense that you meet certain goals. If a goal is met or is accomplished, you observe in a time window, and there you have the DAE tuples, 3-tuples, capture these atomic execution paths. Then you can define the situation. Situation defined to be this 3-tuples and the human desire means the person has a goal; has to have a goal to satisfy human desire. And they have the activity, observe activity, observe the context and see clear goals met. And this is my definition of situation and it is completely computational. Computer computation in the sense that you can have a way to infer human desire. And how you infer human desire turns out to be a big challenge, again. But nowadays, as lots of research again, when you look outside of software engineering, you can see there are people doing other things which can be useful for software engineering research. So I looked into the people doing affective computing, for example, at M.I.T., they can actually infer emotion. And then I did some work in tracking for human desire, which turns out to be also very interesting and I would say, it will take me I think another five years before I can report to you and make some breakthrough. [Laughs.]

Yost: It's a fascinating area of research.

Chang: Yes. And the people working that bring computer interface turns out to be very useful because the hidden mental state, perhaps you can understand from the brainwave. The people working in affective computing, emotion detection is by a study of brainwaves and other things like face analysis and gesture recognition. They can all be very useful for the inference of human desire. If I could solve that problem, then I have a situation well defined, and then I can define intention, human intention. Intention, again,

that's a hidden mental state; but intention in my definition is a path. It's a path, or temporal sequence of situations; so intention is that you have certain goal you'd like to get accomplished or you have certain desire then make sure that desire is met. That's how I define situation and intention, and I infer intention during the software evolution in my paper.

Yost: Can you talk about your decade-plus leadership of the computer science department here and what you saw as the primary goals and initiatives to advance the department?

Chang: Yes. So now, back to this department. Actually, before I came here I studied this department; I think it has great potential. Very collegial department. I don't see any downside to coming here as chair. It's just because people are so nice here. And then I also proposed to the provost and the dean, at the time we need to add faculty. Computer science is becoming more and more important to Iowa State to help fulfill its mission. They accepted my proposal. I actually, during the last 11 years, I believe I hired a dozen new faculty, some with a joint appointment. We actually had faculty in some of the major areas. Software engineering actually grew a lot during my term and we now have a software engineering undergraduate degree, which is a joint administered program with the engineering college. The program has now grown from zero to about 300 students in three years, so that is tremendous growth. When the student enrollment dropped initially in computer science, in those years also we actually didn't fear too much because we were also growing, adding a software engineering degree program. So we are okay, as far as enrollment. Budget cuts are a big hurdle to my plan but fortunately, we are able to recruit very, very bright, mostly young minds as assistant professors. During my term I think owing to my own credit, I would say, we have six more career awardees from NSF. So, totally we have eight NSF career awardees; in terms of our size, which is pretty phenomenal. It's probably the highest density in the entire university here. Again, people are real collegial; they create a very good environment for these young faculty to develop. And then in recent years, because of budget cuts and these things, we did not grow as fast. Last year, during my last year as chair, we hired four new faculty. That's the first time in

the recent five years finally we started growing again. But the department as a whole came a long way from 18 tenure/tenure track full time member to what is now is 28, I think. So it's a good growth. Still, we can see the emerging areas such as big data, we're still short handed. So I believe in the next year, next few years, the new department chair, the new leadership in the department will go into that area. So overall I'm pretty happy with the growth in this department, with the support I got from the different levels of administration, the dean and provost and president are all very nice. The first president I worked with him on a few things, including the JVA Initiative. That's something else I can tell you more about it.

Yost: Please do.

Chang: Of course. Professor John Vincent Atanasoff was a professor of physics back here, way back in the 1930s. the story goes; actually, it was not a story at all before I came here and I began to look into the history and talk to people, read books, and even the court case.. In 1939, he started having this concept of building a calculating machine, because he was very bothered with solving those engineer equations that take months for a very complex one. So you could go to a site, which I constructed. I believe this site is nowadays probably the most authoritative website for because if you Google Atanasoff and go to some other site, they often have a link back to this site, which is maintained by my department. So, 1939 through 1941, he built a machine in the physics building, which is about 300 feet from here. The story goes like this; Dr. Mauchly actually visited this campus, Ames, back in 1941, summer, and stayed in his house, the family, and also visited his lab; and later, there was the birth of the ENIAC. And the rest you can read. Read the website; read the news; read all kind of resource.

Yost: There's a lot of controversy on what Mauchly might have taken from that visit.

Chang: Yes. I just read one book. I did see the letter Mauchly wrote to Atanasoff. That's very revealing, before I know anything about this. But without going into the controversy, there was a fear that Atanasoff had not been really; had not really received his due

recognition until, of course, later after the court case. Many years later, he was awarded the National Medal of Engineering, I believe, by President Bush or the one before that; I don't quite remember. I think he was; he received some recognition in the end but the family, as a whole, never profited from that invention. So I think the Computer Society in particular, probably should also at least make some recognition of his work. And of course, he was a professor at Iowa State so I started this initiative. I talked to President Geoffroy, and then I got involved in serving on this JVA Initiative Committee, Geoffroy's an honorary chair, and there is also John Atanasoff II, the son of JVA. So it turned into something to give him due recognition. And the university was very supportive of this. We almost got this endowment set up and for various reasons it did not materialize, but we keep on working on this. It's going to be another major medal to honor someone who made a major contribution to the computing industry and also one that benefits human beings. We believe that this electronic computer really helped human beings move into the new era.

Yost: And recognizing this important history and heritage, obviously.

Chang: Right. There could always be some kind of controversy, different negative stories, but I think it's all; it's a great mind to be able to in 1941 to be able to come up with this [digital computing] machine [Atanasoff-Berry Computer, or ABC] is just amazing.

Yost: Yes. It really is. Finally, are there any topics I haven't asked you about that you'd like to cover before we close this interview?

Chang: Well, I kept thinking all these years, during my career, volunteer and service career, all these years, many people they really had a big impact and influence and had a confidence in me. I really want to give the recognition and my appreciation to all of them. Of course, if I name someone, I will forget someone else. [Laughs.] But during my term as the EIC of *Software*, I had three associate EICs. Bill Everett from Bell Lab; and then Al Davis from University of Colorado – Colorado Springs. Later he became the EIC as

my successor. And, of course, Pei Hsia from UT Arlington. They were all very, a very tremendous help to me in serving my duties as EIC. And then I have all of these very supportive leaders of the Society who gave me the opportunity to grow; Doris Carver, Leonard Tripp, Willis King, Ben Wah, and all other presidents following me; and then I appreciate in particular from 2007 to 2010, I was EIC *Computer*; I don't recall anyone serving as the EIC for two major publications of the society. The society put a lot of confidence and trust in me, so I appreciate the society as a whole for that opportunity. Then the only major event I'm leading now is the COMPSAC is planning to expand to a major conference. Because in the old days there was only two conferences in the general area. One is COMPSAC and the other is CompCon. One is software and the other is communication. So CompCon has been long gone since the mid-1990s but COMPSAC stays robust for all researchers and is always under the society. And it has served its function in the conference domain, just like *Computer* serves our members in the publication domain. So I think that's going to take me another two, three years' effort to officially establish it as a society flagship conference. After that I can retire. [Laughs.] Not serious, but of course, I would stay active and continue to write papers, proposals, and publish.

Yost: Thank you so much. This has been extremely helpful to document both your important research, as well as your important service in leadership of the IEEE Computer Society.

Chang: Jeff, I appreciate you coming all the way from Minnesota and this is just a great opportunity for refreshing my recollection of what I did for the society, but the one thing I would say is that for all the volunteers, when you get a job, just do your best because the Computer Society is a world class organization. Just do a world class job.

Yost: Yes, I have found it to be a wonderful place and I've enjoyed my interactions with everyone within it, terrific staff and many great volunteers.

Chang: Thank you.