

LASEWICZ: This is an oral history interview with Dr. Diane Pozefsky, IBM Fellow, on July 31, 2003, conducted by IBM Corporate Archivist, Paul Lasewicz.

Thank you, we appreciate the time that you can devote to this. Can you describe your current position and title and basically summarize what you do?

POZEFSKY: Right now, I'm on faculty loan at University of North Carolina at Chapel Hill in the department of computer science. So, I'm a visiting professor there. And I'm currently working on getting some courses ready for the fall.

Specifically, what I'm looking at is some classes that are more industrially oriented than a lot of the research courses. So, the one that I'm working on building right now is on software testing and quality, an area where the schools are basically pretty weak on. It turns out that the testing community within IBM already has built curriculum and so I'm working with them to build off of what they've already done.

And the other course is one that's using one of our major product lines, WebSphere, and teaching the kids how to build complete systems all the way back through the DB2 database

and up front through WebSphere. So, those are the two courses I'm specifically working on.

One of the two initiatives that I'm looking at is the women in computer science issue. Okay, this has been an ongoing problem for a number of years, the number of women in computer science keeps declining. And we're trying to figure out why and what we can do about it.

And one of the things that has come across in a lot of the work we've done, the research and so on, is that women, especially young women coming into the university, are scared of computer science not because of the science and the math as much as the social aspects that they look at. They look and they see very focused, dedicated people who are what they see as single minded on computer science. And that is unappealing to a 19 year old, especially a 19 year old woman.

So, it appears that one of the ways that we need to get to these women is to a) show them that you can do computer science and still have a life, and the second thing is to show them that the applications of computer science can be enhanced if you understand computer science. So, they want to save the world, and showing them that they can save the world better with computers if they understand another level of depth than just writing applications is the key issue.

And the second initiative I'm working on comes back to the curriculum issue. UNC, where I am, is very research oriented. It has a really fine Ph.D. program. When you look at what Ph.D.s do, Ph.D.s do three things: They go to universities to teach, they go into industry in a research lab environment, or they go into a development area. And 90 percent of what you want to teach somebody, maybe 95 percent is the same in all of those three environments. But there's a portion that's different in each one. And so one of the things we're trying to understand and to work on is what are those differences. And if they're small enough and useful enough, and I think they are useful enough, they should be taught to all Ph.D. students because they would be useful.

A simple example of that I'm doing some research with a couple of people, one in networking, one in mobile computing and one of the things I constantly am pushing them on is this notion that says, you [have] got to look at the economics of your research. Are you building something that does give you a five percent increase in value at a 20 percent increase in cost? Just not worthwhile. And if they don't understand that and they don't have that concept built into their research, they're doing research which really isn't going to go anywhere. So, there's a value to understanding the industrial side of it even if you're not [in industry]. So even from a Ph.D. point of view, I think

it's really important. But the area that worries me more is Masters' and Bachelors' [programs].

Computer science has changed radically in the last 10 to 15 years. The ACM curriculum, for example, hasn't been revised since then. And 10 years ago, 15 years ago we were teaching people how to build the infrastructure, compilers, operating systems. I can count on one hand the number of companies now that build compilers. And operating systems is about two.

So given the change in computer science, we really need to start working with the universities and other companies in the industry because no university wants to do what IBM thinks. They're happy to do what the industry thinks. So we need to start getting them thinking about what computer science does today. And of course the really big area in all of this is services. The industry had a minor amount of people with degrees going into services 10, 15 years ago. Whereas today it's 50 percent of hires going into services because it is 50 percent of the business.

So, how do we teach them how to do services? What are the key characteristics we need to teach? And there are simple things like architectures. One way you get intellectual capital transferred between people in the services environment is based on architectures.

And architectures aren't taught anymore in the computer science curriculum. Or they're taught very cursorily. And so people sort of look at an architecture and they don't know what to do with it. They don't understand how it applies and relates to designs and implementations and how to adapt and what the key characteristics of an architecture are. So, there's a whole lot of stuff like that. And so it's sort of time for a revisit and a refresh of curriculum.

LASEWICZ: Interesting. Going back to your background, can you talk a little bit about the school and the education that you had and what courses interested you the most and why?

POZEFSKY: Yes, I was in high school a few years back, when math and science for women really was rare. And so that most of my classes through high school, there would be one or two other women max[imum] in my classes. And so I grew up in an environment like that, which is sort of different from kids in high school today where the classes are more even. And so I had a couple of really good math teachers. I loved math in high school. I'd say that was really my focus as opposed to science. You know, math clubs, math competitions and all that good stuff. It was a game, it was fun, I enjoyed it.

So, when I went to college somehow computers interested me. They intrigued me, I had no idea what they were. But I went into the applied math department at Brown University, which is where the computer science department was. It was just an immediate gravitation. I just sort of grabbed right to it. Again, I had a wonderful professor, Andries Van Dam, who later became chair of the department of computer science, when it became a department and is now actually the vice president of research at Brown. He brought an enthusiasm and an interest and an excitement and just charged up everybody. He's got an incredible record of having brought in a huge number of really top people into the computer science field. So, I guess that it was really that that triggered my interest in computer science.

When I finished my undergraduate degree, I worked for a couple of years before I went back to graduate school and I wasn't really 100 percent sure that I was going to stay in computers. I too had this concern that the people were very focused and that it was a very sterile field and I wasn't sure that I was going to have enough people contact, which looking back was pretty humorous considering that's what everybody does all the time.

But, I worked for a highway safety research center. And this was in the seventies, the early seventies, and it was amazing to me how little computer science and computer

capabilities had infiltrated the rest of the world. So, one of the guys who was a recent Ph.D. who was working there asked me to build a program that allowed him to interactively build tables for motorcycle accidents. And it was a really trivial application. But it was so new and so exciting to that field that we ended up taking it to national conferences and showing people what you could do with it.

And that sort of set a light bulb off for me that said, you know, there's just a huge amount of growth here. A huge amount of opportunity to get this out and help other people. And I think it was that combined with the fun and the enjoyment I get out of playing with computers that got me back into computer science.

LASEWICZ: Could you describe some of your early experiences with science and technology? I mean, you talked a little bit about the work-related example where you built the table. Was there anything else as a young child or early on in your academic career that kind of really said, this is it for me?

POZEFSKY: In high school, I can barely remember the science side of things. So, they clearly did not really click with me. I remember taking the chemistry class and so on, but nothing really. But math on the other hand was, I

think, where the excitement was. And again I'll come back to fact that while I can't remember any names now, I do remember that there was a math teacher who pushed me into taking some advanced courses. This was before AP time, so they can just sort of upped you to the next level of things and got me excited about the math competitions that were going on, which was - if you're not athletic - one of the few opportunities that you had to get out and compete against other schools and meet other people and things like. So, it had a lot of appeal.

So, I think that the competitive challenge and work and meeting other people and discovering that even in a field like that, there really was a social aspect to it, was a key driver.

I talked about the highway safety and during graduate school, again, it was individuals. My advisor, a couple of other teachers who just gave the spark and I'm not sure I can point to specific activities. I was not a tinkerer and I grew up before there was such a thing as home PCs and things like that. So, those weren't the things that drove me.

LASEWICZ: During the course of your career, have you had a mentor and if so, what role did he or she play for you?

POZEFSKY: I never had a person who was dubbed mentor until recently when we had to write it down. But the reality is, yes, I had mentors. And I think probably the two people who strike me the most as having been mentors were Jim Gray who is an IBM Fellow, who is retired a number of years now. And a gentleman named Glenn Huff who was one of the technical leads within architecture and is also now retired.

And it was really interesting to have the two of them because I had the two of them basically at the same time. I worked for Jim for a number of stretches. I worked with Glenn but not for [him]. In fact, I'm not sure he was ever a manager while I knew him.

And there were a couple of things that were wonderful contrasts between the two. Glenn was a technical perfectionist, purist I guess is a better word. And was focused on doing the right technology. Whereas, Jim was much more interested, no, not much more interested, but much more apt to be willing to adapt the technical solution to the political environment or other aspects.

And so sort of learning from the two of them to look at the deep technical decisions and what the right answers are and then understanding that you still had to balance that with

the reality of products and architectures and economics and politics was a really good balance for me.

I think I learned a lot by dealing with the two of them. Because I worked for Jim he had the opportunity to make sure I got the right job. So, I guess in some ways that's not called a mentor if you work for somebody. But in fact it was a relatively small architecture group. And so the ability for him to help me, direct me to the right projects was key to growing my success. And my being able to go to both of them with a technical discussion that said, here are the options and here are the pros and the cons I can see. And having them help me push my way through the next level of detail, tell me how to analyze them and to judge them. And in architecture we talk about the fact that a lot of the good architects have taste, good taste. That is, you can't learn good taste, you just are born with it.

That's really not true because they taught me good taste. And then you just do that by lots of practice and lots of times going over, looking at things and after a while, you say, this doesn't look right, does it? And he'd say no. And then you got to the place where you were comfortable doing it yourself.

LASEWICZ: You mentioned their role in helping you identify the right jobs. Can you encapsulate your work history, kind of summarize the types of jobs you've had?

POZEFSKY: Yes. I joined IBM in the SNA architecture group. SNA was the networking architecture that was the proprietary one that IBM had. It precedes TCP/IP.

It was what was out there in all companies' networks. It actually is still out there. We don't talk about it very much any more, it's not sexy. But, in fact, it is still a very important technology for many of our customers. The world is moving to TCP but we still make a good living off of SNA.

And one of the interesting things was that when we came into that organization, there were four of us who started the same week. We were the first four external hires that had ever come into that organization. It had always been an organization that promoted people from the development organization, the products that actually implemented the architecture. So, this was sort of the grand experiment for the architecture group. And it turns out there were a lot of people who were very against it. They didn't think that was the right way to bring an architect in, they should come up through the ranks. So, it was a real challenge.

And the other thing that was very challenging about it was networking wasn't taught in the universities. So, I came in sort of knowing that computers could communicate to each other, but not much more about networking. So, I started sort of at ground zero on networking, as we all did and learned on the job, which was a wonderful experience.

What was interesting was my dissertation was in formal languages and compiler areas. And so the first thing I was put on doing was working on doing a compiler that we had for the specification of the architecture so we had an executable architecture. And that quickly morphed into taking ownership for one of the components, growing those components and fixing problems with it as we went along.

From there, I started working on some of the new architectures. And the sort of big first step on that one was we were trying to meet key characteristic of the competition. The competition at the time was not TCP but DecNet out of DEC, Digital Equipment. And we were looking to make our architecture and our network more dynamic, more adaptable and more applicable to the smaller computers that were coming up. So, we worked a lot with the Rochester labs, which built the System/38, System/36 and that family, to adapt SNA to the new environment. So, it was called Advanced Peer to Peer Network, APPN. And Jim Gray and I were the leads of that architecture specification, which was

a radical change in the architecture and again, we were looking at it originally for small products.

And one of the most interesting experiences we had was that there was an effort going on [with] the mainframe side with VTAM and NCP (which runs on the 3745, the network box that fronted the mainframe) to become more dynamic. And this had been a requirement coming from our customers for about 10 years. And they were doing a different design to make their network more dynamic. And so we went to them and said, we think that the design for these smaller products would scale up for you. And so let's start looking at that.

And this was a fairly radical notion that said these little machines and these big machines could actually run the same networking architecture without any problems. And so we spent a lot of time going through the politics and the technical details to make sure it worked. And basically got to a point where I actually got involved with the implementation of it and was one of the lead designers for the first implementation that went on there. So, that was a pretty interesting piece of work and a lot of fun.

And as I was finishing up on that is when I took a leave when my daughter was born. And I worked part time for a couple of years while I was doing that. And as I was just sort of coming off of that, it became clear that TCP was a

major challenge to SNA. It was becoming reasonably clear that we were in a battle for our lives and probably going to lose it.

So, we started looking to what [were] our customer's issues and what were we going to do to help them. And it was clear that as they migrated and as they were in this transition period, they were going to have both. They were going to have both networks running in their system and they were going to have applications that were designed for one or the other.

And so I started working on the design of an architecture to address that problem, to allow applications designed for one network to run on the other one. And we called it AnyNet; we didn't have any acronyms for it . We just came up with a name for it.

And as we're doing that, I had a small team of architects that I was working with and it became clear as we proceeded that things had changed in the world and nobody wanted a paper architecture anymore. So, I looked at these five or six folks on the team and I said, when was the last time you programmed? And one guy said, well I did a little FORTRAN in undergraduate. And none of us had been programming for years. And many of us didn't have a whole lot of programming experience in any case anyway.

So, I said, well, if this is going to happen, we've got to get something up and running. So, I turned this team into a development team. It was only a prototype so it didn't have to be perfect and we could take some shortcuts and everything. And while we were working on that, Boeing came along and said, we have a problem. And it turns out someone who heard about the Boeing problem knew about what we were doing and said, why don't you go talk to them.

So we hooked up with the customer and we gave them code to run and to play with, it was not product level code or anything like that. So, basically I had taken this project from architecture through a prototype to a customer activity and then it became a product.

We convinced everyone that it was important enough. We really wanted to get it into other products. We didn't want to be a product on our own. [But] we couldn't make that happen. So we decided to be a product on our own. And so there was a small team developed. We referred to it as a mini product because it was not the same size as all the other products in Raleigh in terms of people. And so I led one of the development teams in that group.

And at the same time we're spending our time talking to the press, to consultants, to other firms, to other customers.

So basically, I took what started as an architecture and went through the entire process to the point where we actually tried to do some marketing kinds of things. Not for the marketing channel, but for the marketing people because we were having trouble explaining to them what the value was.

And so we produced things that were sort of crude but it explained how they could explain if you will. So it was really soup to nuts... Working all the way through getting to work on all of the pieces.

And that was a wonderful experience. Seeing a project move all the way through all those phases really teaches you something about the process and the company that you really can't see any other way.

And as I was coming off of AnyNet, that was when I became a Fellow, I looked around for what were the key things to do, and this was seven or eight years ago. And what we saw was that networking was too hard. Whether it was going to be TCP or SNA, people needed what we refer to today as plug and play. And so I pulled together a small team of people to start looking at these views. And we looked at printers. This was before we had some of the capabilities we have inside the company today and it was before Windows was quite

as good as it is about plug and play capabilities. How do you find the printers that are available?

And we also looked at things like, as people moved their laptops from room to room when there was a problem with them, the IT shops couldn't find out where the problem was. They knew who was doing something but they didn't know where they plugged in.

Then we looked at what we referred to as smart wall plugs. And we had some really interesting neat stuff going on. But I wasn't getting anywhere with this. [I] worked for Al Zollar at the time in networking. And I went to him one day and I said, we've got to fix ease of use. And I said, but the reality is SNA is on the downward slope and getting people excited about this and interested about this doesn't really seem to be happening. Then he looked at me and he said, "kind of like putting lipstick on a pig." I knew it was time to move on.

This was about a year after Atlanta, the Atlanta Olympics, which as people remember was not one of our sterling moments. We had a lot of problems with it. And the Nagano Olympics were coming up and a friend of mine who recently retired, Ed Lassetre, had been asked to take a look at the Olympics and what some of the problems were and what we could do better. And as I looked at it and talked to him

about it I said, it would be sort of interesting to get to the other side of the network to actually roll one out and deploy one and see what happens.

So, I joined the Nagano Olympics for a year. Right after I joined the group, I went to Nagano and they had a big sign with a countdown for the number of days, it was 365 days. So, it was a year before the Olympics that I started working on that.

And that was a wonderful experience in many ways and in many aspects. One of the most interesting things was learning to deal with people from a very different culture. And the Japanese culture has come a long way.

When I was...it was 20 years ago because I was pregnant at the time and my daughter is 19 years old now. I was going to go over to Japan to try to convince them to use the APPN technology on some of their office systems. And several people said, don't bother. They won't listen to you because you're a woman and the culture there 20 years ago was such that it just won't work. And my husband who is also an IBMer was supposed to go to England to give a paper on another architecture he was working on, which I was pretty comfortable with.

So we looked at each other and we swapped. So I went to give the paper in England and he went to Japan. But he failed anyway. So, I'm not sure it mattered which of us went.

When I went to the Olympics it was clear that there were a number of issues. There was clearly at that point in time, and I think it's gotten better, but at that time, there was a discomfort between the American and the Japanese parts of IBM.

And part of it was that the Japanese sort of felt we were coming in and trying to tell them how to do it. And they felt they could do it. And I think that there's a whole lot of history in some of these organizations.

So, out of the year I spent leading up to the Olympics, it took me about six months to establish my position. And I think part of the issue was I had a position and title that was above the people I was working with but I was a woman. And they didn't quite know how to deal with that. And of course, there were all these language issues. But, you know, once I could show them that there was a value we could add and once they understood that I wasn't trying to run the show or do it, but I was trying to help them, we made progress. If they had a security issue, we'd find the right security people and bring them in. If they had a problem

with configuring one of our products, we'd bring that in. They were having problems with products. We had some really interesting phone calls; when you deal with Japan, West Coast, East Coast and France, . there is no good time to have all of those people on a phone call. And so it took us about six months to sort of establish ourselves. And then things just ran beautifully.

They understood what we were bringing, what we could do and we understood what they could do and it was a wonderful partnership. One of the really neat experiences. And the Olympics itself, just being there and working with it was just truly incredible.

I need to tell one of my stories there. We had what they called the Surf Shack, which was a marketing project where the athletes had Internet access and Fan Mail, where people could respond and talk to the athletes and everything. And it was very successful.

The Olympics started and at the Curling Venue, which was off in the middle of nowhere, the athletes didn't have access to the Internet because the Surf Shack was too far away. So, they decided they were going to set up a satellite.

And one of the things that we had the athletes do was make sweatshirts with their picture on it and their home page and

stuff. And so anyway, I ended up running the satellite place there. And so I was the Internet lady.

And there were a couple of things. One was the curling community are true amateur athletes in the sense of the Olympics. They are amateurs, they have very little support from anyplace. So, they learned to use the Internet better than anyone else at the Olympics.

The judges there and the coaches typically had family at home who were involved with curling and they were in touch because that's where the newsletters were, that's where the information was, everything was posted on Web sites. They were amongst the earliest ones to do that.

They were communicating back home through the Internet and so I was asking somebody why and they said, I paid for this myself, I can't afford phone calls from Japan to Norway. I can't afford this. The Internet is free.

So, they had rapidly learned to do that and they were just really passionate about their sport and responded to sometimes rather weird fan mail. But they always got around to answering every mail and I mean, they just loved it.

And after when we closed it down because curling was one of the earlier sports, I'd occasionally bump into them on the

train or something and it was old home week. They were just a bunch of wonderful, wonderful people. So that was a wonderful experience.

So after the Olympics, after a year and a half of traveling overseas because I worked a little bit on the Sydney Olympics after that, it was tough. It was time to move on again. And then [I] went to work for the CIO's office.

And basically they were concerned that they had networking problems. Well the reality is by the time I got there, they really solved most of the networking problems. They still had a lot of other problems. But the ones that they were worried about had diminished; they had a really good group of people doing it. So we did a lot of fire fighting in terms of going to specific sites and trying to analyze why they were having a problem and coming up with some recommendations for it.

One of the really big issues that was happening right around that time is as we kept acquiring companies, particularly in IGS. So what we had was a huge number of independent networks because we bought a network when we bought these small companies.

And Canada was really struggling with that, trying to consolidate. And the acquisitions are something we're still

doing. And we will probably continue to do for a long, long time, especially in Global Services -- where we didn't even hear about most of the acquisitions. I mean, they're small, they're a specific company. When they out source we pick up those groups and things like that. So that's a fairly major piece of work that's still ongoing. But that certainly was a really interesting piece of the activity.

But as I said, most of the problems were really solved there. So I sort of looked around and said, where are the real issues? And one of the things I became concerned about was mobile and remote workers.

And so I sort of started focusing on those areas. And as the company becomes more widespread and we have more distributed teams, we have started depending on a lot of technologies in terms of getting people to work together remotely much more. We don't travel as much and so on. And in many cases, it wasn't that the technology was a problem. The technologies were available, but the rules were different for everybody. People didn't know what the rules were. And so we had a lot of hygiene to do to fix up some of those things. So different divisions had different rules about what the employees could buy and stuff like that. And that stuff was sort of easy to pick up.

And then there were some technologies that we needed to accelerate the use of and move a little faster on. But the biggest issue it turned out was all cultural and people. I mean from simple things such as when you have people on the phone. The mixed meetings were really where the problem was: we had six or seven people in the room meeting and then a couple of people hooked on the phone. And you would forget to make sure to include them in the conversation. And we tried to work on some of the rules and characteristics of how to make that work.

At the time, the staff meetings we had were a mix of people in the room and people on the phone and Steve Ward was the CIO at the time. And I said to Steve one day, I said, if you really want to make them more effective, don't go into the room when you have most of your staff, which of course, wasn't going to happen. It's just not the way people work. But, I said the reality is if the main person that's being spoken to is sitting in the room, then they will speak to you. If you're on the phone, then they'll speak to the phone.

And people on the phone, will make sure that you know what's going on and things like that. So it works much better. But that's just sort of an example of the issues that we had with remote workers and mobile workers. And some of it has gotten better and some of it hasn't.

I've worked Women in Technology conferences for a number of years; we'll have our fourth one next year. And that's run basically by a group of volunteers who don't know each other, don't see each other, who are all across the country. We try to get people from outside the country, but it's basically a U.S.-based organizing committee.

And it worked incredibly well because nobody was sitting in a room with anybody else. We all were on the phone. And so you had a pure phone culture on there. And it was sort of an interesting experience having worked with these folks for a year and never having seen some of them or met some of them.

I walked into the room one year when I was doing program chair and my co-chair was a woman I had never met. I walked into the room and got there a little late and didn't recognize a lot of people. And all of a sudden, I recognized the voice. And it was Janet Warton. I said, I'm sorry Janet, that can't be you, I have an image of you that I've had for a year and you don't look like that.

But the point was that because we had a culture and an arrangement where there was nothing to make it not work, working remotely worked just fine. And it's sort of nice to have an example of the fact that you can do it very well

and very successfully, to remind you when you get frustrated at things. When you're sitting there on a phone call and somebody just won't let you interrupt. And you're sitting there banging on the phone saying, I want to talk. It is possible to make it work and it's sort of nice to have some proof points.

So, let's see. I was working for the CIO's office and I got a call that they needed an architect over in the new storage area, for storage networking. And I did that and I went over to work on that.

That sort of disappeared because they moved out of Raleigh. And as it moved out of Raleigh, I went to work for Lotus until I took this job for faculty loan.

LASEWICZ: Well, as you look back over the history of your jobs from position to position, are there any common themes that you find linking them? Are there any things about the positions that were there that you found attractive?

POZEFSKY: A large number of them were architecture positions. Okay, so there were a few pieces that weren't. But most of them were architecture positions. And I think what's attractive to me about an architecture position is it's sort of , as defined within IBM I should say, but at the intersection of research and development.

Typically the architecture group is the group that is trying to take things forward, invent new things, make significant progress. And that's sort of the research aspect of it. But you're not working on a green field. You're dealing with products on a regular basis trying to make things better and make them be able to move forward. And it's that sort of intersection and that combination that I find the most challenging.

LASEWICZ: Would you say that's the thing that you find most satisfying too, is resolving those issues? Or are there other things about these kinds of positions that you enjoy?

POZEFSKY: The fact that you get to deal with so many different people and so many different products I think is a lot of fun. But I think it is recognizing that you can make significant changes without breaking what we already do.

One of the things that you see in the standards bodies is that as they move from a release to a release, they don't always worry about that. And in the Web area, the standards have been really terrible about doing that.

So, you move up to the next release and you have to rewrite your code, it is very frustrating for both products and

customers to have that disconnect every time. And so the ability to make revolutionary changes in an evolutionary manner I think is the piece that is the most satisfying.

LASEWICZ: I would say that you've had a pretty extensive academic background. As you went through your work career, were the academics relevant? Do they remain relevant today?

POZEFSKY: When people ask me about curriculum I say, the only thing I want you to really teach the kids is how to learn. There is no chance that what you would teach them at the university today is going to be relevant 10 years from now if all you do is teach them technologies.

I mean 10 years ago, they were teaching them C. Not a lot of people writing in C anymore. So teaching them specifics and teaching them the specifics is not very useful.

What is critical is teaching them the underlying principles. When I learned to program, we started with literally machine code. I mean, my first assignment was ones and zeros. And then we moved up to assembler code. And it was second semester before we went up to a higher level language.

And they do the reverse now. They teach the higher level languages and then considerably later, in second year or so,

they'll get around to just assembler code and things like that.

And I don't know that the order that you learn them is important. But what is important is that you can recognize what it's doing and what the implications are. So, we used to joke that what you really want is compilers to put certain things in red, sort of with flashing letters saying this is really expensive guys, do you realize what you did?

And with optimizations in compilers today they can sort of pull it out of inner loops and everything. But the reality is is that as we go to more abstract models for programming, understanding what is complex and what isn't, if you don't understand the underlying capabilities, is really hard to understand.

I think Web services is a wonderful example of this. People sort of assume that Web services, it's like a function call, you can do it. But the reality is it's a huge amount of overhead associated with going to another machine, sending data to that other machine, processing things and come back with. Grid computing use is another kind of example, anything that's distributed programming.

And if you don't understand what you're doing with that, you're going to make some foolish mistakes in terms of the

granularity of the functions that you do. And we see that coming out of some of the standards work that's happening. That we're constantly fighting with the standards bodies who are trying to do things that are not going to scale well because they're just going to be too expensive.

So understanding that underlying capability becomes that critical piece. So, it seems to me and certainly what's been useful for me is that the underlying concepts of what computers do and how they do them really hasn't changed over the years.

How we as programmers and users access them and go after them has changed radically. And I'll come back to user interface in a minute. But if you don't understand what it's actually doing, it's really hard to make good design decisions.

So, it's the fundamentals that are critical. Once you understand how to program and how to write an algorithm and how to design an application, what language you do it in becomes secondary.

I mean, one of the early lessons I learned was they gave us a very complicated formula, this is as I said when we were still writing ones and zeros to make it work. And it turns out that if you analyze this formula, which if you didn't

analyze it, it was a very long program to write, but it actually simplified down to three X plus two or something like that. Huge simplification.

So if you didn't invest up front to understand what you were doing and to look at it, you spent a huge amount of time programming something that you didn't need to do. And that lesson has stayed with me forever.

Analyze first, design first, spend your energy thinking about what you're doing before you start jumping in and coding. And the coding is the fun part. Getting to see something run is where the immediate gratification [is], and we're all into immediate gratification, right? But if you spend that extra time and that extra energy, you make your life easier and everything goes smoother. So that was a real important lesson.

The user interface and the application piece [are areas] where we're doing a terrible job of educating people. There are some really good people who are working in this area. And the basic thing that you have to look at is what is the function that the user is trying to do. So user centered design is basically what the term had been and now we're focused on user engineering, which is a broader more engineering-oriented approach. But the notion is to look at tasks, look at what people are doing and based on analyzing

that, then you can make decisions about the user interface. If you just try to make principles about the user interface, it might be a wonderful interface but if it doesn't match the actions and the activities a person does, it just doesn't happen.

But this is still an art. It is not a science yet. And so it's behind some of the other parts of computer science, and so that's going to happen.

And I think that's one of the most interesting things I see happening in computer science is, we now call it computer science software engineering and we think of it as a discipline and as an engineering discipline. Whereas, 20 years ago, we thought of it as an art.

When people introduced things like structured programming, I was a graduate student at the time and so we all sort of pulled out our code first time the article came out. We all pulled out code that we thought was good code or that other people had written that looked like good code. And we discovered that we'd been writing structured programming for a long time. And what we realized was that a lot of the technology and a lot of these changes that were coming out, and I think is still a true statement is trying to take the average programmer and make them a good or better programmer as opposed to taking the people who are already good or

excellent programmers and do something to them. Because they, in fact, intuitively understood the discipline and used it.

So, I mean, we have to remember that when a lot of the things that are proposed for processes and technologies are doing that. And the reality is average by definition is the majority of the people. If everybody was an excellent programmer, then that would be average. So, it's sort of by definition that life is a bell curve. And the goal is to improve the quality and capabilities of the big chunk of people and that's where we sort of need to remember what we're doing.

One of the areas, for example, is as we speed up our development process how do we make sure that we maintain quality and get things right. We've just run a couple of studies and Bill Woodworth who works in testing and software quality just sort of has been pulling all of this together.

The reality is that as we try to meet the needs of the market, we are in danger of forgetting our processes and forgetting our roots and what has been successful over the years.

You think that you can speed things up by skipping a step. And you find that, in fact, you just pay for it later. And

when you pay for it later, it always costs more. Lots and lots of study statistics showing that. So, trying to get the discipline back and still meeting the market needs is the real challenge we have today.

LASEWICZ: Okay, this question deals with more cultural aspects. You had talked about your experiences in dealing with Japan and your experiences you had as a woman.

Can you talk a little bit about some of the changes that you've seen in the course of your career at IBM, having come into the technology field as a woman in high school without a lot of ladies in your classes?

POZEFSKY: The first time I went on a business trip to a conference, it was up in New York, I don't remember where it was. I think it was at Yorktown, but I wouldn't swear to it. I walked into the room and it was over 100 men and there was me. And maybe one minority person in the room. I mean, it was an all white male room. And that doesn't happen anymore. It may, on occasion. But it doesn't happen nearly as much anymore.

We've done a huge job at bringing the diversity in. And I think that it's made [a difference]. I mean, for all of the reasons that diversity is valuable in terms of getting the right skills and getting the right people, but also in terms

of getting different view points. We've clearly made a huge, huge step up in that direction.

The reality is I think I've been really fortunate throughout my career in that I've had managers in organizations and teams that have been very good about dealing with women and minorities and so on.

So, when I look back there were very few times I had problems with it. There was one gentleman I worked with who.... There were three of us working on a project. I was sort of managing it and there were two technical people. And the key technical person was a woman and he always called the other person -- the guy. And when we were sitting in a room, he always talked to the guy. And I was very aware of it. It was bothering me fiercely. And I made an observation about it.

And a couple of the guys who worked for me started watching it too. And they saw it as well. And, of course, this became sort of a humorous for us because we knew where the work was getting done and the capabilities.

But what the guy started doing, who was getting all of the questions when we were meeting, is he would look away. So, if the person who was being somewhat sexist here was trying to ask him a question, if you're looking away and down it's

really hard to have the conversation with the guy. And so he was sort of forced to talk directly to the woman. And so, those things still happen. And they still exist. But you sort of try and find ways to deal with it without looking at the guy and going, you're a sexist pig. It's just not a good thing to do.

I say try not to do that. I have dealt with some people who it was very clear that they weren't listening to me. And so you find a way to get to them. And I tend not to be confrontational, so I try to find other ways.

And a non-gender issue, for example, there was one guy I worked with who was in a key position and wasn't really good at listening to other people. And he had to adopt the idea as his own before it would take place.

And so we sort of...several of us were struggling with this and we finally realized that if we went in and had a friendly chat and said, you know, I've been thinking about and this is sort of what we're doing. We did this for a week or two and all of a sudden it was his idea. And he had grabbed onto it and owned it himself. And so what happened and so...and in many ways that that's the way you have to deal with people independent of gender, you deal with them that way. You kind of find what works.

There are other people who have different styles, who are willing to be more confrontational. It doesn't work for me. And I think one of the things that has actually been helpful over the years in [my] career is the fact that I'm not a confrontational person. And that's probably somewhat gender based. A lot of women aren't. And finding the way to work with people and finding different ways to make those connections and to get your point across I think has helped a lot.

I have a friend who said that she got tired of always being put in a position of being the peacemaker and the arbitrator and facilitator. But that's what she did really well. And so you'd end up with her and six guys who were ready to beat chests against each other. And her role was to get them to talk to each other. Well there were guys who would do that well too. So, the diversity issue and the gender issue are about styles.

I mean the reality is more women just tend to have this style rather than this style and more men have this one and this one. But what you really want when you're looking to build a team is the diversity of styles to get them to work well.

There are other reasons that you want diversity of gender in culture because things are different and people work

differently. Any time we deal with people in Europe, we learn something new in the language.

I was in Hursley [England] talking to folks and they were having a real hard time with this design that I was talking to them about. And one of the things we talked about were bots. Okay, to me, this was a normal computer science term. Everybody talks about bots on the Web.

And after about 15 minutes, one guy goes to me, he said, a bot here means a baby's bottom. Okay, so I said, okay. I guess we'll have to find another term. And things like calendar versus diary. The simple things with language actually get in your way. I mean if you don't get the diversity of organizations and countries, you lose things like that.

There's a story that one of the early Think Pad designs where they had the keys set to open the top of it, turns out it worked great for males, but it didn't work for anybody with fingernails because you couldn't get your finger there because the fingernail was in the way because it was turned too much. And they found this out the first time that a secretary went to use it because there were no women in their design team, or at least there were no fingernails. They may have had women in there, but none with fingernails.

So, I mean really simple things like that. The size of people's hands. There are some things that are going to work really well for people with big hands or little hands, but not the other. And in terms of cultural things, when you go to make a trip, to set up a trip for yourself, people do things differently. And if you have a very homogeneous organization or group looking at this design, you're all going to do it the same way. And so all of a sudden you start getting the breadth of people in there and the different cultures and things like that in there, you start recognizing the differences.

So, we need to get those pieces in there. And we're doing a whole lot better. You know, are the numbers as good as they should be, no. Will they ever be? I don't know.

There are a whole lot of issues in today's society that it's really hard for IBM to try to change. Of course, one of the biggest ones is that the pipeline isn't there. And we keep on looking where we need to get to change things. And I'm very involved in National Engineers Week in North Carolina. We've actually got a coalition across the state of engineering societies, other companies, universities and the department of public instruction for the state trying to get to our more rural areas.

But we're focused on middle school because the statistics show this is when the girls start veering off and stop taking math and science. And there's lots and lots of statistics in there.

On the other hand, I remember looking at things going on in my daughter's preschool where you started seeing gender separations. Which toys they used and things like that. And I was really, I worked hard to make sure my daughter had the truck, as well as the doll, which she didn't like either of them. So, you win some, you lose some, right?

But, no matter how much people tried, there is something going on in our society which is clearly pushing it because it starts so young. And maybe part of it is nature as opposed to nurture.

But it seems to be something that we're teaching the kids really, really young. That they're picking up from society, from television, from language, from role models, I don't know. So we're focused at middle school because that is from an educational point of view where we see [it]. But I believe that we have some fundamental issues that are going to be a whole lot harder than that to look at.

Of course, one of the issues is that engineering and science still aren't considered cool. How many TV programs are out

there where the science and the technology and the engineering are really stars? We've seen them over the years. MacGyver, there was even the A-Team to some extent was a way of dealing with mechanical things. But we don't see a whole lot of that anymore. And none of those had women in them.

So, there's lots of things. We have Ally McBeal for lawyers and things like that. The first Women in Technology Conference that we held, we had a photographer there who was a guy and he was one of the few guys, needless to say. And he talked to me afterwards and he said, you need a 'got milk' campaign. I looked at him. He said you need something as catchy as the 'got milk' and little white mustaches to tell people that science and engineering are cool.

And there was a National Academy of either science or engineering summit on women a few years back. And the same thing came up. How do we get the message across that science and engineering are cool? Because if we get the message across that it's cool, it will go after both guys and girls.

There's a discussion I had about the application versus the deeper depth in computer science. It is not just women who have this problem. You see lots of guys who don't want to

get in because they can't see the value of getting further in.

The goal is to find the fundamental issue that is key and some of those you'll find that they apply more to women than to men, girls to boys, but they are basically gender independent. It is a personality issue, it is a perception issue that is causing that difference as opposed to gender. So, I think that all of that is really important to get across.

Just another observation that came out of I think MIT or Carnegie Mellon, I'm getting some of my reports mixed up. One of the things they observed is that all of the things that we talk about the women in computer science seem to be for U.S. women at U.S. schools. And when you look at non-U.S. women at U.S. schools, none of it applies. It looks like it's radically different. And there's lots of guesses as to why it is, but fundamentally, we believe that the women who are coming in from other countries to go to school here in computer science or in engineering have a different motivation and background as to why they're here and what their goals are than the U.S. women.

So, as we worry about global issues, we need to observe now that most of the research and most of the concerns that are going on have been very U.S. centric. We just did a piece

of work within IBM within the academy about women in EMEA (Europe, Mid-East and Africa). And one of the follow on pieces of work is when they have some panels within the U.S. and AP take a look at and say which are the things that apply and don't. My own reaction was it was 90 percent the same. But there are a couple of things that were radically different and in EMEA based on laws to a large extent. Well, in some places based on law, some based on culture.

For example, one of the things that while they're on leave or working part time, there are a lot of country rules about what they can and can't do. So, it's a whole lot easier to have somebody in the U.S. working part time while they're on leave and get them access to things like that. Whereas there are things that actually prevent you from doing that in parts of EMEA. And Germany and France have very strict labor laws that are interfering. And the whole part time model has really not taken off. And I'm sure it's related to the laws and things like that.

So, that was one of the areas where there was a difference. But the vast majority of things we saw were very, very similar across the two. I mean their education system is different than ours so you have different leverage points perhaps as to how you want to fix them. But the observations are all the same. It's a pyramid. Drop, drop, drop out.

LASEWICZ: You've had a successful career at IBM since you joined in 1979, had to overcome challenges and obstacles. One of them was work life balance. Can you talk a little bit about how you've tried to manage that over the course of your career?

POZEFSKY: A few years back my daughter and husband bought me a sweatshirt that said, stress, what happens when your gut says no and your mouth says, of course, I'd love to. Some of us do better than others on work life balance.

But one of the things is when we decided to have children, my husband and I talked about it, and I concluded that we could deal with one child. If I started having to balance two children, along with two careers, it was more than I could cope with. If we had one career and two children, I might be able to deal with it.

So, the first thing was recognizing that we all have limitations and that these limitations are to a large extent very personal. Because I have friends who deal with five kids, I can't imagine how they deal with five kids, I would die. Never mind any careers. You have to understand your limitations and recognize what they are.

The other thing was making a decision as to what was important. And I've walked out of meetings with vice presidents because I had to get home. Of course, some of those were hard-stop constraints. When your daughter is in day care and they close at 5:30 and you have to pick up your child by 5:30, you walk out of a meeting at 5:00 because it doesn't matter what it is.

I worked with teams and people who understood this and they knew I was serious about it and I didn't say, I've got to hard stop at 5:00 and stay until 6:00. I also wasn't irrational about it and unreasonable. It wasn't like I said, I can only work for two hours and things like that.

So that by establishing that I was serious about it and being reasonable about it, people accepted it. And once the sort of guidelines and rules got settled down, they were there.

On my calendar at work, I always had my daughter's soccer games on. So if she had a game on Tuesday afternoon, a home game and I wanted to be at it, it was blocked. And people saw that and they recognized it. Now, it was before we had the little ability to put private on it there [the calendar] so you couldn't see it, see exactly what it was. So people knew that this was an important issue for me and that I was going to be there for my daughter.

So if you establish those principles, what I found was people are accepting. Again, as long as you don't say, well, I have to leave at noon everyday because...no one is going to accept that. But if you set reasonable assumptions and reasonable rules and are willing to put in the time to make sure that you get the work done when you need to get the work done, what I learned was people were very accepting of it.

In terms of actual balance, I do a lousy job because I tend to take on too many things. What I found was when my daughter got into junior high, high school kind of time frame, I had to start cutting back on community activities.

And in some senses it's very natural because in schools especially. The parent involvement at the elementary school is very large and drops off as you go on. So, there's sort of the child doesn't really want it, the school doesn't really want it and as your career causes problems, then it turns out to be sort of a natural timing thing.

But the principle that I try to set is that to be reasonable about what the balance is. My daughter complains that vacation always includes computers. And I happen to be an early morning person, my husband and daughter aren't. So I can get up and work for two hours before they get up on

vacation. And that sort of solves problems and we can take care of things like that. My husband obviously will stay up later because he's more of an evening person. And then, of course, teenage girls, they don't have any concept of time.

So, you try to balance things like that. But establishing the ground rules and making the decision and trying to be rational about it is clearly what it is. But, you know, there's the old cliché. I never met a man on his deathbed who said, I didn't give enough time to my company.

My concern with one of the things that I see is on work life balance is people focus so much on people with children as the issue. And we've gotten better at recognizing that fathers have a role with children as well. But people who are single, people who don't have children still have a life. And I was at one site where one of the women said, "one of the problems I have is I'm single and people just assume that I can work any hour, any shift because I'm single". And there was some upper level managers in the meeting when she said this. And I saw them squirm but I hope it got fixed. Because everyone has a life.

And in fact, one of the biggest issues coming for me at this point is while my daughter is in college now but I have parents that I need to deal with and they live in the Boston area, I live in North Carolina. Dealing with parents on a

remote basis as they get older and they're both in a nursing home at this point and so at least I'm not worried about their day to day safety, but there's all sorts of issues you have to deal with.

And everybody has those issues, and they're not going to go away. In fact, they're going to get worse over years as people live longer, the notion of having to balance children and parents is going to get worse. And we have to address it.

And what I see often is our policies say it's there. But they're implemented very unevenly within the corporation because to a large extent it depends on a first line manager. And if you've got a good first line manager who recognizes this and is comfortable with it, that's one thing. Then you'll have another one who doesn't. So at roundtables, I consistently find very, very inconsistent balance between the two of how people do it.

One of the things that has improved a lot is people are much more comfortable with people who are telecommuting or teleworking, I guess is the more politically correct term. There are managers for a huge amount of time who said, I can't see you, I can't see your working, how do I know you're working at home? And I don't hear that as much any more.

I mean people have started recognizing that if you have a call at 7:00 in the morning, it's a whole lot easier to take it at home. And if you have a lot of calls that commute time is not only wasteful, but sometimes you can't even squeeze it in. So, people have become much more comfortable with working at home.

Now I tried working at home at various times. And it didn't work for me. And it didn't work for me for a couple of reasons. And one was I missed the socialization. I like being around other people.

And the other one is that there are too many distractions of things that have to be done around the house and it just drove me nuts. I sat there with, you know, I really should do the dishes. So just getting out of the house was really important to me.

And so one of the things I tell people is the first thing you have to decide is if you're somebody who can work at home. And not everybody can work at home. I met somebody recently who started working at home and they said they had a huge amount of trouble getting started in the morning. Once they got started, they were okay, but getting started was really a killer. So what they ended up doing was they went out and got themselves a cup of coffee at Starbuck's

and then came home to work. And they said that routine, getting out, getting started and coming back and sitting down and working made the difference for them.

So you've got to find out what works for you. And there's all these things that they tell you about have a dedicated place and things like that, which is important for some people and just isn't an issue for others.

But, try and set down rules and say this is going to work for everybody, isn't going to make it. It's very, very personality dependent as to what you can do and what you can't do. So, coming back to the work life issue is those are the kind of calls you have to make and figure out what works for you and what doesn't.

But the other thing is if you're unhappy and you're worried about other things and you're not doing things with your family that are causing problems for you, you're not focused at work. So, you've got to figure out that says, I'm not happy with what's going on, I've got to go talk to my daughter's teacher and I'll be able to focus on work for the next two weeks. Whereas, if I don't do that, it's going to be nagging in the back of my head and I'm not going to really be as focused.

So, you go ahead and you take the time off and you make that happen and you do what you need to do. I remember having a meeting once where my daughter, I can't remember if the school was closed or she was sick or something like that, but she couldn't go to preschool or school or something. And I had a meeting I had to be at. I was running the meeting, so it was really bad. And I had people coming in from out of town and everything. So, I brought her in. She was little at the time. I gave her crayons and her coloring book and said, just sit and I'll bring you cookies every once in a while. I mean she was in the room with us and I could get away with it. But it was the only thing that was going to work for me that day. Would I do that every day? Good Lord, no. But, some flexibility has to be in there and my daughter understood the flexibility and my husband was very flexible.

One of the things that I tell people is that when I look at most successful women, and there are a few exceptions, they have some sort of very strong support basis to work from. Whether it is a spouse or a significant other or parents or children or somebody, they almost always have somebody that they can depend on. I mean, with every rule there's an exception. And there's this one woman who says, I have made myself as independent as possible. She goes, I don't even have a plant that I have to water. So that I can be where I want, when I want and not worry about these things. So,

there are women like that. But to a large extent, that support and I shouldn't say that she has no support, she clearly has friends and things like that, but she's sort of taken a different tack on it. But to a large extent you need somebody sort of supporting you so that to sort of make you have that balance.

One of the things that I tell people is we buy season tickets for plays and things. Why is that important? Well, I paid for the ticket, it's on my calendar, I'm going to go. If I had to made a decision last minute to buy, it's too easy to say, no, I'm too busy, I'm not going to do that.

So, again, it's recognizing your own style and your own personality and recognizing that making that commitment at the beginning of the year for the next 12 months for those tickets made it easier for me. And I have other people who schedule dinner with folks or schedule lunch with folks or schedule coffee with folks because if they don't schedule it, it just won't happen. And so you've got to find the ways that will work for you and force yourself to do that balance.

LASEWICZ: As you look back over your career in technology, what would you consider to be your most significant contributions?

POZEFSKY: The two technologies that we worked in the architecture group, APPN, which was going from a static to dynamic architecture within SNA, and AnyNet.

Neither of them today when you look at the big picture of life, are that significant, but they were [significant], at the time, critical to getting people to move forward and to progress. But those were sort of my two biggies.

LASEWICZ: What are the qualities that you consider have led to your success?

POZEFSKY: I think that there's some technical ones and some social ones. A technical point of view I think that one of the capabilities is to be able to analyze a problem and figure out how to apply either technologies or research into that environment. So, the reality is there's very little that you create from whole cloth. Occasionally, there are people who truly invent a new algorithm and things like that. But the vast majority of work is adapting and applying whether it's a technology or an algorithm or a product.

And so, being able to analyze a problem and find the pieces that other people have invented, I mean, we all stand on the shoulders of others. I think that that's one of my technical strengths.

The other technical strength is an ability to simplify. People can come up with solutions for things that are terribly complex. And if you stick with the terribly complex, it's never really going to make it.

And so, I'm never happy until I've simplified it down to as simple as it can be. It tends to get more complex over time as everything does. But if I can't get down to something simple, I don't think I have the right solution. So, I'll keep pushing and pushing to get things as simple as I can. So, the keep it simple stupid principle I think is my ability to help do that, I think is my other technical strength.

From a social point of view, people point of view, I guess, social is not quite the right word, is recognizing when people aren't communicating and being able to bridge that gap. And the second is to recognize what's important to people. And what's sort of going to make the sale.

I learned this lesson one day when I was presenting to Share, which was the user group originally for the 390, but it's grown since then. And I was presenting something. And the guy in the front row was really giving me a hard time. And it turned into a dialogue after a while. And finally, I said something about how it's going to help IBM. And he

goes, I got it now. Now I believe this is real. If I couldn't see how it was going to help you, I didn't believe that you were going to be interested in this technology. So, he had to understand it is win/win because he wasn't going to buy it if he thought we were claiming to be altruistic. And sort of it was one of those things that the light bulb goes off because you've got to find out what this person's principles are , how they decide if something is real or not, and hit that point. Because it doesn't matter what you think. If you're trying to sell somebody on something you've got to find out what their point is. And I think that that's one of my skills that is good is being able to dig out what the principles are for other people. I like to think I'm logical as well, but I'm not always sure about that.

LASEWICZ: Well, that's the last of the question I have. I know we've covered a lot of ground. Is there anything that you'd like to add?

POZEFSKY: I can't think of another thing.

LASEWICZ: Again, thank you very much.

POZEFSKY: Sure.

[END OF INTERVIEW]