Margaret Burnett Oral History Interview, January 10, 2017

Title
“Striving for Gender Equity in Computer Science”

Date
January 10, 2017

Location
Valley Library, Oregon State University.

Summary
In the interview, Burnett details her upbringing in Illinois, including her early facility for mathematics and the role that her parents played in cultivating this talent. She then outlines her undergraduate years at Miami University of Ohio, noting her studies in Mathematics and, in particular, her earliest exposure to the discipline of Computer Science.

Next, Burnett reflects upon her initial professional job at Procter & Gamble Ivorydale, which hired her as it's first female manager. She then describes the seven years that she spent as a private sector employee and entrepreneur while living in Santa Fe, New Mexico, and likewise details the circumstances that led her and her husband to relocate to Lawrence, Kansas. From there, Burnett traces her master's and doctoral studies at the University of Kansas; recalls her earliest exposure to the Internet; and touches upon her founding of the Lawrence Women's Network.

The interview then turns its attention to Burnett's career at Oregon State University. In this, Burnett responds to questions concerning her early impressions of the university and the Computer Science department; her research on visual programming languages and end-user software engineering; the development and impact of the GenderMag protocol; and her contributions to information foraging theory. She likewise touches upon her collaborations with OSU researchers working on machine learning; comments on her associations with Hewlett-Packard, IBM, Microsoft, and Saturday Academy; and shares the background behind her strong advocacy of undergraduate research opportunities. The interview concludes with Burnett's thoughts on the contemporary status of women working in the technology sector; her perspective on change within the College of Engineering; and her ideas on OSU's positioning as it looks toward its sesquicentennial anniversary.

Interviewee
Margaret Burnett

Interviewer
Chris Petersen

Website
http://scarc.library.oregonstate.edu/oh150/burnett/

PDF Created February 11, 2020

Transcript

Chris Petersen: Ok, today is January 10th 2017 and we're in the Valley Library with Margaret Burnett who is a Distinguished Professor of Computer Science at OSU, and we'll talk to her a fair amount about her career in academia and elsewhere, but I'd like to begin by talking about your early years and ask you where you were born?

Margaret Burnett: Yeah, I was born in Springfield, Illinois. I had a lot of family there that - my extended family was there actually, because I was probably the third generation in my family to live there. We had a department store, my extended family. Most of my uncles and cousins worked in that department store, although my father didn't; he was a lawyer. And I spent my whole childhood there.

CP: That's interesting, the department store. I'm wondering if that was an important place for you growing up.

MB: Yeah, it was actually. So of course, since we were a department store family, that was the only place we were allowed to shop. So occasionally, especially in my teenage years, this raised angst because, you know, the in brand would only be available at some other store. And so anyway, there was that.

Let's see, I worked at the department store on vacations and Christmas holidays and that sort of thing, and that was actually really fun. I had a huge variety of temporary jobs. So I was a gift wrapper in some of the Christmas seasons. My sister, my younger sister, ended up doing that too sometimes. I would be an occasional substitute salesperson if somebody got sick, so I did that from time to time. Let's see here, I was an assistant to some office staff. So we were in charge of payroll and lots of other business-y things. And so I spent a summer being an assistant to them and that gave me a lot of variety of jobs. And it was around those super old buildings - nowadays they are preserved, I guess, as historic buildings. But it had been sort of cobbled together from neighboring buildings. And so the floors were at different levels, and there were secret staircases leading between these buildings back behind the nice pretty display areas, and there was still a hand-cranked elevator back in the freight room. And so somebody would actually have to operate it. And yeah, it was a fun place to be doing vacation jobs and that kind of thing.

CP: What sorts of things were you interested in growing up?

MB: Well, I've always been kind of a puzzle-worker. I've always liked math, so anything that involved problem solving I was really into. And a lot of times that was math, or things like cryptograms or stuff like that. I loved those kinds of things. And my dad also had been pretty good at math, and so occasionally I'd be having trouble with math and so I'd go to him for help. And in his wonderful, gentle way he would help me and I would just get so mad because I hadn't been able to get it and he had. Which is of course crazy, I was the kid right? [laughs] But I really wanted to be able to solve everything by myself; that was kind of a great fun thing for me.

But I also really liked theater, so I was in the junior community theater things at various moments in my childhood. And I loved music and dance. My mom was kind of a semi-classical pianist. I mean, she didn't do it as a career, but she just loved to do it, and mostly what she liked to play was classical and opera music. So we had a lot of music around us as we were growing up. And I took a lot of years of ballet lessons and tap lessons and I liked all of that.

My mom had really liked the theater too, and when she was a girl she grew up in Chicago where they have an opera. So back then what they would do is they would allow high schoolers and up to be extras in these opera scenes that required crowds. And so she liked to do that, so she would go after school and go be in crowd scenes. And one time she was a little slow - the crowd was supposed to exit - she was a little slow in getting off the stage, so they closed the door before she had actually exited. So she was stuck on stage. And so one of the leading singers saw her back there stuck, so he went back there and got her and put his arm around her like she supposed to be there and whispered to her "for God's sake, sing!" And so that was her big theatrical moment, I think.

But anyways - so I did all that stuff. I wasn't really talented at any of it though. Mostly I got theatrical roles because I could memorize the lines. The same with dancing roles; I really didn't have a lot of talent. But it was fun to do. So anyways...

[0:05:09]

CP: So the early interest in math, do you think that sources to your father? Or where do you think that came from?

MB: It's kind of interesting, isn't it, to think about that? Probably. But my mom was quite a problem solver too. So, she majored in chemistry when she was in college, but she didn't make it all the way through; she got sick. Probably it's something we would now diagnose as mononucleosis, but at the time they didn't know what it was, so she just sort of slept away a year. And that's kind of the end of her college career. But when growing up, she also was very mechanical, so she liked to fix things around the house that broke. My dad was not mechanical at all, so she would do that stuff. So I think I probably did grow up in a family of problem solvers.

Yeah, and actually my grandmother – you know how those old fashioned drinking fashions go straight up with bubbles? And nowadays they don't, they go slanty? Well, my grandmother was actually the one who invented that slanty thing. She was a graduate student and discovered that these up and down ones are bouncing the germs up on the top, and so it was helpful to prevent spreading disease by having a sideways one. But she didn't get credit for it or patent it or anything like that. This was, I think, the university at that time owned the intellectual property for it. So yeah, always a family of problem solvers.

CP: It's interesting to hear you talk about these women in your life, generations before you that were operating in environments that weren't necessarily very friendly to women.

MB: Yes, that's true. And my grandmother, she still had very traditional values. Growing up, when she was trying to figure something out, my mom would talk about how she made it seem to be her husband's idea. So she was very traditional in her ways of thinking about gender roles and that sort of thing.

But my mother was a little peculiar. She wanted to be traditional, but at the same time she was awfully feisty. And so I think a lot of her life was sort of struggling between those two values. At some point when she was in college, hypnotism was all the rage. So this would've been, what? I guess the '40s. And so they would go to these parties and try to hypnotize each other. And so, you know, they are doing the watch thing, you are getting very sleepy, very sleepy. And she's trying to get very sleepy, very sleepy, and kind of half succeeding. And then they'd say something like they would tell her she can't do something, you know, "you can't lift your arm." And she'd snap right out of it and say "oh yes I can!" And so she was never one to go along with "you can't do something." And this also applied to gender roles. "You can't do this because you're a woman" and "you can't do that because you're a woman." And so she kind of forgot to tell me that there were things I wasn't supposed to want to do. Maybe that's one reason my interest in math survived longer than some other women of that generation did.

CP: I'd be interested in knowing more about your school experience in Springfield. Did you feel there were obstacles being placed in your way as you pursued this interest?

MB: In school, I don't think there were obstacles. At least not obstacles that any of us noticed. So no, I don't think so. I feel the school system I went to was really good. They had what, at the time, we called tracked classes. And so you could kind of place yourself, with the advice of your parents and your school counselors and that sort of thing, in whatever level was kind of the right level for your own interest and ability. And so we had like B track English, and B track math, and A track math and this sort of thing. I think there were about three tracks of most of the different main subjects. So I always got myself into the A track math because I was really interested in math and, I don't know, more B track history and stuff like that. So that was really nice. And I feel like I got an excellent education, and everything I wanted to do was encouraged by the teachers.

But on the other hand, there was society around me. And so it was definitely not a good idea socially for a girl to be good at math or anything academic for that matter. I still remember the "it must be nice to be so smart." [laughs] But the biggest obstacle was really one that wasn't placed by the school system, it was kind of, again, a society thing.

So I couldn't figure out what I could do with math. And so, you know, I'm getting to the end of my high school years, I was expecting to go on to college. Most people in my family did. But I knew I didn't want to be a teacher, at least at that time I knew that. Because from what I had seen of math teachers or any other kind of teachers, mostly what it was, was discipline, and that didn't sound very enjoyable to me. So I just couldn't figure out what I could do. It was the only career I could think of to do with math as a woman, because it was the only one I had ever seen.

So there was my next door neighbor, who was eight years older than me, and she too was very good with math. And she went to college, majored in math, and graduated and got a job with this company called IBM. And in that company she was what was called, I believe, a systems analyst at that time. So basically it was computer science. So I thought "hooray, a career! And it's not teaching!" So I decided that's what I was going to do. I guess I would've decided to be a computer science major in college if such a thing existed, but at the time there was only one university in the whole country that had that as a major. There were several that had it as a minor, but only one that had it as a major. So I chose that university to go to, partly because my cousin went there and invited me out there for a weekend and I really enjoyed it. And the year I went was the first year they actually offered computer science as a major, but I decided to be a math major and a computer science minor.

So that university was Miami of Ohio. And so I walked in there never having seen a computer. Nobody had ever seen a computer. They were these things the size of buildings and nobody ever got near them. And of course, I'd never used one.

CP: That's fascinating to hear that Miami of Ohio was the first school to offer computer science.

MB: Yes It was, at least as a major.

[0:12:08]

CP: Interesting. Well, tell me more about your progression through Miami as a math major and budding computer scientist.

MB: Well yeah, so Miami was another very good educational experience, I had a wonderful time there and really learned a lot. And so I was a math major, and that was in the College of Liberal Arts and Sciences. So we had quite a broad curriculum that we needed to satisfy which, I believe, was called the Core Curriculum there. So I loved that, because the variety of learning was something that was just, I mean I had always thrived on a lot of different subjects. So I thought that was awesome.

The math classes turned out to be a little more theoretical that I actually liked. I was more of an applied sort of mathematician and the program there was much more theoretical, and so mostly what we were doing was proofs. And so that sort of frustrated me. I could do them, but I wasn't getting the same enjoyment that I had gotten out of high school math.

But I loved computer science. I mean, it was like a puzzle, so every time there would be a new assignment, man, I'd just race home and start working on it. This would be so much fun. So it just totally grabbed me and I did well in it. Back in those days, the way you did computer science assignments is you stood in line – well, first you wrote everything out by hand, you wrote out your whole program by hand. Usually we used these special sheets of paper for it so that you could tell what column everything would be in, but sometimes you could get away without that, as long as you were really careful when you were typing it in. Then we'd stand in line in a special room and eventually get to sit down at a key-punch machine and punch up all our cards. And then we'd turn them in, hopefully we wouldn't drop them, because that was bad. Really bad. Then we'd turn them in, into the window, and they would take them in and, you know, some mysterious process would happen that we didn't get access to. And if we were lucky we could pick them up the next day. So it was quite different than today's trial and error environment; you tried really hard to get it right the first time and not make typographical errors. And if there were logic errors, then it's not like we had a debugging tool or anything like that to experiment. All we got were these core dumps, which were these gigantic pages of hexadecimal that you had to decode if you were trying to debug using that. We did, or else we could insert print statements and try again. But this, of course, would be another day gone by. So that's what debugging was like.

And so, in order to get faster turnaround or not have to wait so long in a line for the key-punch machine, we started keeping weird hours. We would go in at 2 or 3 in the morning or something because that was the only way we could get decent turnaround. So I used to joke with my friends that the reason so many computer science people are thought of as weird is that our training made us weird. [laughs] We eventually had to do those things.

But anyway, yeah, it was a lot of fun and by the time I graduated I realized the right major would have been computer science with math as the minor, and then I would've been able to get an extra year of computer science courses. But by then I was in love and eager to go join my soon-to-be husband and get married and start our life there. So I didn't switch my major.

[0:15:49]

CP: It sounds like it was a pretty supportive environment the whole way through.

MB: Yeah, it was. It was very good. And on the very last day I needed some form of signature. So this was, I think, after final exams and everything, the faculty were still there. And I went around to get this signature from, I don't know, maybe it was a math professor or maybe it was a computer science professor, and he said "what? You're not going to grad school?" And grad school had never occurred to me, nobody had ever mentioned it. I'd seen grad students, but it had never really been something that had been broached to me and I just didn't know why I'd ever do it. But I had enjoyed being a student so much that the thought kind of stayed with me and I thought, "hmm, you know maybe this is something I'll do someday." So that one little sentence probably had a huge impact on things that happened down the road.

CP: Yeah, you eventually did. But there is a gap - a nine-year gap.

MB: That's right. Well, more like seven, because I needed time to actually get the degree. Let's see here - so we move on, right?

CP: Sure.

MB: Yeah. So then I got a job in industry. At the time there was a shortage of computer science people, as has been true in many ages including right now. And so I got a lot of interviews, and I ended up getting hired by Procter & Gamble. Miami of Ohio was in Oxford, Ohio, and Proctor & Gamble Ivorydale was the place I was hired, and that was in Cincinnati, not too far away. My husband had already graduated and he was teaching down there, so this was perfect.

And so Proctor & Gamble Ivorydale is this large complex. I think at the time it had 13,000 employees; it's much larger now. And it involved a lot of manufacturing plants mostly clustered together in this Ivorydale section of Cincinnati, but also some in other places in Ohio. Anyway, the Ivorydale division had never had a woman software developer. In fact, they never had a woman in management at all, ever. And so I became the first one. And the reason I got that job, even though I was a woman, is that my boss fought tooth and nail to get me. So it was, I think, his only choices were me or this guy, and the guy was elsewhere in Proctor & Gamble and wanted to transfer into that, and he had no background at all. And my boss did not want him. But he had to fight and fight to get a woman in; it was just unheard of to have a woman in the manufacturing environment at a management level position. But they hired me.

So here I was, I guess I was probably – was I 21 yet? Yeah, 21. So here I am, sort of paving the way, although I didn't think of it that way. I just thought of it as "this is my first job." But it was kind of awkward because, for example, who do I go to lunch with? Am I supposed to go to lunch with the secretaries? Am I supposed to go to lunch with my peers? Is that awkward? It's like me and all these guys. So there were things like that. And then, when I would go to these meetings, people didn't quite know what to do with me either. And so they wanted to be welcoming, so they would try in their ways to be welcoming, but they didn't exactly know how because people didn't know nearly as much about being inclusive as we know today. So they would say things like "Oh, a bright spot for this meeting!" And this all made me feel very uncomfortable, but I didn't quite understand why at the time because we just didn't know as much as we know today. And I knew they were trying their best to be welcoming, so I just didn't really even understand why I was so uncomfortable. But it was with things like that.

Anyway, it was so weird, I got to meet the president of the company my first year there because, you know, I was unique; the only one in management. But by the time I left a year and a half later, there were four of us women at the management level in Proctor & Gamble Ivorydale. So that changed things a bit. And now when I look on the web, I can see that Proctor & Gamble is one of the leaders in gender inclusiveness and other forms of inclusiveness too. So they certainly, at some point, really got religion about it and took off with it.

[0:21:11]

CP: So what came next after a year and a half?

MB: Well, I have quite a variety of things I have done in my life. My husband had always wanted to go out west, this was a grand dream of his, and so that kind of had been part of the deal all along. Now, I really liked my job at Proctor & Gamble, and I'm kind of a sink in your roots kind of person, so I wasn't really excited about going out west after I'd come to enjoy Proctor & Gamble so much. But a deal's a deal.

So we took this vacation, I guess it was about a ten-day vacation, to sort of explore the West. So we were young and foolish, and so we flew to Albuquerque, rented a car, and made our way by car through several places up to Salt Lake City and then flew home from there. And this was our scouting out a place to move to trip. We fell in love with Santa Fe, so we picked that. And at the time I tried to sort of interview around there and see if I might be able to land a position, and I thought I had one lined up; it looked very promising.

So we went back with kind of a secure feeling that at least one of us would have a job out there and, oh I guess about three months later, we made our move. That was actually an adventure in itself. So we rented a U-Haul. My husband went out again, he drove the car out and flew home, and he rented an apartment and got it all set up. And then we packed up a U-Haul over 4th of July weekend and started to make our way out there. And it was so hot. And we were driving from Ohio through the Midwest through super hot, super humid territory. I mean, it was like 100 degrees and like 100% humidity. It was so hot and muggy. And back then, only the most luxury cars had air conditioning and the U-Haul truck certainly didn't, and it was generating heat of its own. I think if we'd been older we'd probably had suffered from heat stroke, we were so hot. I remember stopping in St. Louis and going into some truck stop that was air conditioned and being too hot to even talk to each other. We just sat there and melted with ice tea for a while before we could even talk.

But anyway, we made it all the way out there to Tucumcari, which is New Mexico, which is not too far from Santa Fe. By this time we had only enough money left to stay in a hotel or eat dinner, but not both. [laughs] And so we booked a room. We were able to afford one person's dinner, so we shared a dinner. We didn't have enough money for breakfast the next day. But Bob, my husband, had set up a bank account out in Santa Fe, so when we finally got there we would be ok, but we weren't ok yet.

And so then, the last morning we drove to our apartment and parked the car. And that was July 3rd or 4th. Parked the U-Haul in front of our new apartment and moved our stuff in and, in the middle of the night, some party-goers came along and slashed all four tires. Oh, and I forgot about the distributor going out somewhere along the way too. Anyways, it was a very adventurous trip, but we made it, and U-Haul dealt with the tires. And all's well that ends well.

Anyways, so then we got there, and it turned out I didn't have that job that I thought I had lined up. So Bob, he quickly lined up a gig helping to dig water wells, just to keep food on the table. He had been an Earth Science teacher, you know, but it's like, "we need food." So he started doing that, and I started looking around to see if there were computer science-y jobs there. Well it was a small community and, no, of course there weren't, other than the state, which was the job I thought I had and I didn't.

But there was this guy starting up this new timesharing thing, which was the kind of business where somebody owns a computer and then other people have terminals to dial into it. And so basically they're paid by the hour. And so these other people would be small business – well, usually slightly larger than small – but local, well-established businesses. Anyway, so this guy was starting up this thing, and they had only two employees; they couldn't afford anybody else. And so they couldn't hire me, but they said "well, there's this accountant, and he's interested in starting to use our services, so maybe he needs somebody like you?" So I went to see the accountant, and he had a number of – he was very forward thinking – and so he thought, "you know, in this day and age we need to start automating stuff. And so I want to use this new timesharing service to do this." And so he had lot of connections in the community and lots of clients, and so this is what he could bring to such a business. And then I had the technical know-how, and so he said "well, let's be partners!" And I said "ok!"

And so, of course, what I didn't realize at the time is if you have a partnership, what you don't get is a salary until the business starts making money. And so I was working now, but still not making a salary. But over time we did start making some money, and at that point my husband decided to go back to school and get a degree in engineering. So I continued doing that, eventually I spun off a business of my own in Santa Fe. Loved it there. My husband commuted down to University of New Mexico and picked up his engineering degree, and then went to work for a couple of engineering companies in Santa Fe. So we stayed there for seven years and loved it.

[0:27:19]

CP: And then came grad school?

MB: Kind of sort of, yeah! So then, by that time I was pregnant with our first child. My husband got recruited away by the USGS, and so that would've been Lawrence, Kansas. And so I thought, "oh, this is going to be a good chance to go to grad school" because the commute down to UNM sounded like more than I wanted to do and we didn't have a grad school in Santa Fe at that time. So I thought, "ok, that's what I'll do." So we moved to Lawrence, Kansas, and I had my first child. I started going to school part time to get a masters in computer science. By the time I graduated – so I was off and on part time – by the time I graduated, our second child had been born. So our two sons were raised in Lawrence. So I got my masters, and then I got together with a professor in Computer Science and started a small business consulting on computer stuff, because by that time people were starting to buy microcomputers to run their businesses.

So, microcomputers were in a terrible state. People were building them in their garages and throwing together some software – it always come with source code, thank heavens – and then they would go out of business. And so then these small businesses would be stuck with this very buggy software and this chunk of hardware, and they wouldn't have any way to make things work. And the company is gone, so they basically needed rescue people. So that's what I did; my friends used to call me the fire department. And so terrible things would happen, and they would call us in, and we would go ploughing through the source code trying to find the bugs and fix things and get them back on track. And so yeah, that was really fun. The name of our business was Bulgrem and Burnett.

And so my husband, he worked for USGS for a little while and then started working for Black and Veatch out of Kansas City instead. So that's what we were doing while we were raising our boys. But once in a while things would happen like the day care center would close, because we'd take the kids to the day care center that was associated with the university, and the university has this habit of closing over breaks and such. And so then there'd be this sort of, you know, "now how are we going to watch over the kids?" So a lot of times I needed to take time off. And so I'd take time off and, of course, I love the kids, but after about four days I'd get really starved for problem-solving activities and adult conversation. So I'd start having these brainstorms.

And so one time I had this brainstorm where I thought, "you know, we need a professional women's networking association of some sort. All these men, they have ways to get together and help together but we women, we don't know each other, we don't know each other exists. I'm really tired of being treated as the tag along wife." So I decided to start this network. So I did, I called every professional women I could find in the yellow pages and tried to interest them and started this thing called the Lawrence Women's Network, and it still exists. It's so cool – it's still there. So anyway, I started that.

Anyway, I was greatly enjoying my job, but consulting has a lot of ups and downs, and so I thought, "I need something to even out the downs." And so since I had a masters I was qualified to be a lecturer at the university, which means a temporary instructor as needed for particular classes. So this gave me what amounted to steady quarter-time work, which would be a nice evening thing. And I discovered that "whoa, I do like teaching after all!" I liked it so much that I started thinking about getting my Ph D.

[0:31:31]

CP: Can you tell me about your early experience of the Internet? I'm guessing it happened sometime around then, with the early versions of email.

MB: Yes. That was about the time that I went back to get my Ph D. I started on that about 1987. So there was email at our university; this was the first time I had experienced it and so, yeah, I'd get emails sometimes as much as two or three times a day, it was really fun! [laughs] Almost always from professors there at the university; I didn't know anybody outside the university to send email. Nobody used it for personal use, it was all professional and the only people who had access to it were universities and government installations and research labs, that's about it. And so yeah, by the time I graduated in '91 with my Ph D., that was how all the universities were going about the job search thing. You were still sending your resumes and all that stuff on paper, but the initial contacts and sometimes the follow-up for references were happening over email now.

CP: How had things changed for you as a student between being a master's student and a doctoral candidate? You're at the same place, but the field is changing so rapidly.

MB: Yeah. So there was about a seven year gap there too between the master's and the Ph D., and because the field was changing so rapidly it was pretty much like all my master's work has gone out of date now. So I almost had to repeat the master's just to be up to speed to be able to do the Ph D. So yeah, a big gap in a rapidly moving field like computer science isn't the best thing between a master's and a Ph D. The gap between a bachelor's and a master's, on the other hand, is kind of awesome. Or at least it was for me because it really helped me to gain perspective about how various things I was learning in academia fit in to real world situations. So if I were doing it all over again with perfect hindsight, I'd keep the gap between the master's and the bachelor's and not have the gap between the masters and the Ph D.

[0:33:52]

CP: What was your focus as a doctoral candidate?

MB: I did visual programming languages, which at the time was a very hot topic. Nowadays, people are very familiar with visual programming languages or visual programming environments, but back then everything was entirely text. And so people were inventing entirely new languages which came with their own visual environments. And Windows for heaven's sake, which was also very new; everything was very new.

So yeah, it was kind of a programming environments project. And when I graduated, I was regarded as a programming languages person because that had been a programming languages-sounding topic. And I remember being surprised to find out that's what I was, and it's like, "oh, that's what I am!" It never occurred to me that I would need to inherit some kind of label that fit in to kind of pre-conceived notions to what everybody was. So, you needed a label. Looking back on it now, I would say it's more like a blend, what I was doing then, of programming languages, software engineering and human-computer interaction. But mostly people didn't think in terms of blends back then. Anyway, so visual programming language was a very hot topic.

At the time that I got my Ph D., it was a bad year to be graduating with a Ph D. in computer science. The market was off and it was darn hard to land a position. Most universities, then and now, don't really hire their own, so we knew we needed to go elsewhere. I ended up getting hired at Michigan Tech, which is way up north in Michigan. So that was a good place, but what they didn't have yet was a Ph D. program in computer science, and I really wanted one of those because I had a lot of momentum in my research and I wanted to have Ph D. students. They did have permission to start one from the university, but it hadn't actually started. We were quite happy there, but it wasn't my first choice because of the lack of a Ph D. program. Furthermore, my husband had never forgotten his love of the West, and so we kind of wanted to be out west again.

And so we had kind of narrowed down to the Pacific Northwest as the area of choice for us, even though neither one of us had ever been there. But nobody in the Pacific Northwest had even given me an interview the first time around. But anyway, then at some oddball time of year here comes this ad from Oregon State across the Internet. So I talked to Bob and we decided well, you know, we're not really on the job market again because this is pretty good. But just that one place, because that was one of our top choices. So I applied and, lo and behold, I got an interview and, lo and behold, I got the job. So we moved out here almost twenty-five years ago and we've been here ever since.

[0:37:04]

CP: So drilling in to your early OSU years a little bit, I'd be interested to know about your memories of your initial impressions of the university and of the town.

MB: Oh yeah. Well, I loved it and loved everything about it, and still do. At the time, similar to today, Corvallis was about 50,000. The university though was about half the size it is right now. I've always grown up in towns that are about 50,000 and lived in them. So other than Cincinnati, every place we've lived has been a small city. So this just felt perfect to us and yet a reasonable drive to Portland if we wanted big city amenities. So yeah, it was perfect for our family goals, it was perfect for everything, the community.

The university was also very supportive. At the time I was hired, I was hired the same year as Cherri Pancake, and we two were the first women who had ever been hired into tenure track positions in Computer Science. There had been some instructors and stuff, but never a real faculty member. And in fact, we were numbers 3 and 4, I believe, of female faculty in the College of Engineering. So our dean, Dean Owen at the time, was very forward-thinking about trying to be more supportive about getting women faculty members in, so he made it a priority. So there we were. [laughs]

The department was really supportive of us. One thing I remember really well is Mike Johnson, he was a senior instructor. He had been with the department a number of years but still didn't have a tenure track position, just because I think that's the way he wanted it. Anyways, we were up in the main office and I was standing by the counter plowing through some student files that I needed access to. So some students came up to the counter and assumed I was the receptionist because I was female and there, and so they asked me some kind of receptionist kind of question. And Mike Johnson came over, the senior instructor, and he said "oh no, this isn't one of our office staff, this is Professor Burnett. But I'm one of the office staff, how can I help you?" And I thought, "what a class act," you know? What a class act. So yeah, the department, the college, they were very supportive all the way along.

[0:39:41]

CP: And in kind of a broader sense, how would you describe the state of the Computer Science department in those early years?

MB: Well, it was much smaller than it is now. I don't know how many faculty they had, eleven maybe? Something like that. A very congenial group, very helpful. So we had Ted Lewis back then, he was pretty well known, he had quite a reputation. And I was trying to figure out how to build my reputation. Oregon State wasn't particularly well known and so I didn't have the nice letterhead to help me climb the ladder. So I decided, "well, it would be great to throw a workshop at one of the major conferences on visual object orientated programming, and then to put together a book based on that." But I thought, "this is never going to sell unless I have somebody famous on board." So I talked to Ted and said "I'll do all the work, but I need somebody famous," and he said "yeah sure, I'll do it." And he said, "I'll contact my friend Adele Goldberg, I bet she'll do it too." And Adele was super famous and she got on board too. So we three threw this workshop and ended up publishing a book on it, a collection of edited chapters, and yeah, that really helped. And perhaps in part because of that, I ended up getting a National Science Foundation Young Investigator Award, which is considered to be extremely prestigious, and that certainly helped to build my reputation too.

CP: Was the computing infrastructure adequate?

MB: Yeah, it was. I never had any complaints; I always had everything that I needed.

CP: Well. tell me a bit about setting up your research program. I'm interested especially in how you developed the point of view of your research. It sounds like it flowed out of your Ph D. work, at least initially.

MB: Yeah, it did. So I was considered a programming languages person when I landed, and so I began to consider myself that as well. And so I started doing – I was doing language design really, as part of that visual programming language thing, and I had a number of questions that I hadn't finished investigating. And so people who program, they think of things like abstraction and exception handling and event programming, those kinds of things. And in my language and in most visual programming languages, we hadn't really considered grown-up language topics like that. And so I tackled those one by one with my students here at OSU. And so I think that went on – well that's pretty much all we did for the first five years.

And really what I was after was something we called the scaling-up problem in the visual programming community. And so the problem was that these visual programming languages were nice little proof of concept things that worked on toy examples, but what about the things you need for programming anything real? And so that's why I was looking at things like abstraction and event handling and those types of things.

We worked on that almost exclusively up until about '96, and then I also started bringing some software engineering things into my research. And this was largely because, at that time, Gregg Rothermel began as a faculty member here, and he was a software engineering person and he and I started collaborating a lot. A third member of our department, Curt Cook, who was an empirical software engineer, also joined that collaboration very soon in the end. So we started looking at things that are outside the, shall we say, "me and the language" kinds of things into the kinds of things software engineers look at.

So just to give you a little bit more context here: programming language research is really about a programmer and creating a program. You think about the language and the kinds of programs people can create with them and that kind of thing, but when you get into other phases of the software lifecycle, like testing and debugging and design and requirements and all those kinds of things, those don't really have to do with the language, they have to do with a whole lot of other aspects of software development. So that's called software engineering. So Gregg and I started looking at testing visual programming languages. He had been researching testing in sort of classic software engineering, and I was the visual programming person, so we put our ideas together and came up with a new approach to testing, especially for interactive visual programming languages, called WYSIWYT: What you see is what you test. And Curt was very very experienced at doing empirical studies that relate to software engineering of how humans actually do this stuff. So he joined our collaboration and began to look at whether actual humans sitting in front of that programming language could actually succeed at using this event.

[0:45:17]

So that was the beginning of something that turned out to be much bigger. Eventually what we came to realize is that we were working in an area that was about software engineering for people who might not be trained in software engineering. So we started expanding our efforts to spreadsheet users, because spreadsheets are actually a form of visual programming language. And so we brought WYSIWYT to the world of spreadsheets, and we started thinking about other aspects of what we called end-user software engineering. So, that phrase is software engineering by people who don't have any software engineering training, might not even have any programming training, but yet they somehow find themselves in these software engineering tasks. But they don't have the training for it and furthermore they don't even wish they had the training for it. And so instead of pretending like they wish they were software engineers we had to say, "well, how can we bring software engineering gains into their work processes and their knowledge sets and their motivations and priorities?" And so it became very human-computing interaction.

At this point I was kind of – I had drifted from programming language design to the intersection of programming language design and software engineering, and now there was this huge human-computer interaction component to it too, so it was really an intersection area. So we coined this term, "end-user software engineering," got some grants, and started some collaborations with some researchers around the country, and formed a consortium called the EUSES consortium, which more or less stands for "end-user software engineering." So we had collaborators at Carnegie Mellon – and, in fact, we still do – University of Washington, University of Nebraska, Cambridge University in the UK, Penn State. Anyway we had acquired quite a few, and then our own local Saturday Academy, so there were also education pieces in it too.

At the time we started this consortium, I think there was something like 140 papers out there in all of the literature that I could find that had much of anything to do with end-user software engineering. And by the time – I can't remember the statistics, something like five years later – the literature base had increased ten-fold. And so now there were people who weren't in our consortium either who were starting to do end-user software engineering, and it started showing up in keywords for major conferences and in granting agencies and that sort of things. So that whole sub area became established. Gregg and Curt were among the people who started it with me, so it wasn't just me. It was a team effort to get this thing off the ground and all those other researchers together. We wrote a grant and NSF got behind it and yeah, it really took off. So let's see here, pause for breath?

[0:48:54]

CP: So the gender work, I believe, started to flow out of this, is that correct?

MB: It did, yes. So let's see here, we started the end-user software engineering stuff big time in about 2000. By that time, most of my students were now working in this end-user software engineering thing instead of language design. And as I said, we got more and more into human-computer interaction too. And so one of my masters/Ph D. students, Laura Beckwith, had just finished her masters doing something end-user software engineering-ish, and she was trying to figure out what she wanted to do her Ph D. topic on. At that time, people were finally starting to talk a little bit about the underrepresentation in computer science, and they were talking about it mostly in the education environment. So Laura and I started brainstorming about whether there were things about software environments that also weren't as gender-inclusive as they should be.

So she decided to take that on as a research topic, and she just caught fire with it. She started reading from five different fields to try to pull together what might be the right hypothesis to go after in that kind of space. So she read from education and from psychology and from the academic discipline of marketing and from computer gaming and from feminist literature and everything she could get her hands on. And so she'd run across these theories that tended to be pretty well established empirically as well, about how people problem solve, about how people learn, and gender differences in those things. And these hypothesis started dropping in her lap. You know, things like, well, if there are these statistical gender differences in say, for example, learning styles, then shouldn't that apply to the way people learn new software features? So she got this raft of hypotheses to go after. Didn't have time to go after all of them of course, but we picked some of them and started doing some experiments in our lab to see if those things from other fields actually did pan out in software.

We kept finding out over and over again that they did. I mean in hindsight it's like, "duh" [laughs] but at the time nobody thought we'd find anything. In fact, I remember when Laura was preparing for her – I guess it was her pre-lim exam, and so she was saying she wanted to do her thesis on this topic, and her committee – which was, of course, all male except for me – said "well, ok dear, you can go after that but you won't find anything." And I was afraid she'd give up. I was afraid that she would think that, "ok, it's too discouraging, I'd never find anything." But at some point after one of those kinds of sessions she turned to me and she said, "you know, they can't mess with my head anymore." She said, "I know so much more than them about this." [laughs]

So anyway, we started finding lots and lots of differences. And I guess the best way to describe it was, it's really all about individual differences, it's not about gender differences as a sort of bucketing method mechanism. If you can imagine some sort of graph about how, let's say for example, about learning styles when you're learning new technology. You can imagine there's this style and that style and maybe the females are like this and you can imagine that maybe the males are like this. And so you see a lot of overlap here, so it's not about women are over there and men are over there; there's a huge amount of overlap. But if you start paying attention to the long tails, you see that a lot more women are on this side of the long tail, a lot more men are on that side of the long tail. So if software is mostly designed and implemented by males, then what that says is it's probably just the overlap plus the male long tail that's probably going to be supported by the software. And so what this is going to mean is that there's a few females that are going to be supported and a lot of females that aren't. And this is what we kept finding statistically, over and over again. Which is, as I say, in hindsight seems obvious, but it wasn't obvious at the time. And pretty soon other people started doing it too, finding the same sorts of things that we had found. So yeah, that was the beginning of our gender HCI effort. Pause for a drink of water.

[0:53:57]

CP: Sure, yeah. So where does the GenderMag method fit into all this?

MB: So, in probably about 2012 I got this email from this business person. And he wrote that his particular software company – well, he wasn't the owner, he was a product manager, but anyway this company he worked with, their target audience was a particular branch of medicine. And as it happens, something like three quarters of that branch of medicine is females, they're the medical practitioners. And he had discovered that females mostly hated his software. So they knew they needed to fix it, but they didn't have any idea how.

And so he started plowing around on the web and he found my name as somebody who had done research in that area. So he contacted us and said, "please help, we don't know what to do." And so by this time Laura had graduated, some other students had followed in her footsteps and also helped us understand some things. But we had never tried to put together anything practical, we just had these papers – studies and papers. So what am I going to do, you know, send this guy a bunch of papers? That isn't going to help him. That's when I realized we needed a practical method. And so Laura started helping us with this also. I didn't really have any funding for it at this point, so we were kind of doing it in hobby mode for a while and it sort of drifted along. But eventually I did get some funding again and it was able to take off in a bigger way. And that's how this new method was born.

It's called GenderMag and it stands for gender inclusiveness magnifier. And what it is, is a method that anyone involved in the creation of software should be able to use, and by "should" I mean "I hope so," and my empirical results are very promising about this. And I also mean that's our design goal, is that anyone involved in the creation of software can use it. So to be a little more concrete, that means managers – product managers who work for companies – software developers who are actually doing the programming, it means software engineers, it means user experience people, design people. Anybody who has their hands in the pot should be able to use it. And it's an inspection method. So what they can do is they pick some path through their software. So let's say it's table software, and let's say the path they want to evaluate is adding a new employee. So they pick something like that – in software engineering this is called a use case – and then they take some prototype of that, maybe they have software that actually runs already or maybe they just have design sketches, but whatever it is, something they can kind of walk through for that use case. And they start evaluating the steps, one at the time, using some of the GenderMag questions and some of the GenderMag materials.

And so, for example, if it was payroll software and they were adding a new employee, maybe the first step was click on "maintain employee files." So this is what the designer would like them to click on, and everybody decides based on some gender data that the GenderMag data provides, whether or not they think that that method is something that people with different learning styles, different problem solving styles, different information processing styles, would choose to do. And you know, there might be a lot of reasons why not; it might be really well hidden. That probably doesn't have to do with gender, it's probably just usability. It might be something that – let's just say its labeled in a foreign language that not all of them can read. So that too probably wouldn't have gender things, it's just usability things.

But it could also be that some people might say that, well, "maintain employees files, really it's not maintenance that I want, it's adding that I want. Maintenance isn't something that I really care to get into, that sounds more advanced than what I want." So people who view themselves at being not particularly proficient at software might not click on that. That's one slice of the population where there could be gender differences.

Another slice of the population is how they might learn it. So they might get into it and it might turn out that there isn't much in the way of information about how you succeed at it. So people who like process-orientated learning might feel sort of left high and dry. People who like poking around and tinkering might not. And so those are just two different learning styles, neither are better or worse, but there are gender differences in how people like to learn about those things.

So anyways GenderMag helps people find potential issues like that for software developers and the software creators, and then they can decide whether they think they want to fix those. So they might say "ok, you know, we found that thing but you know, we don't really see that as a high priority thing." That's their choice, GenderMag helps them, but it doesn't tell them how they should prioritize things, that's up to them. But if they find something and say "aww man, this is never going to work for people who like to learn by process, but I know how to fix it," then they just add it to their bug list or features list, and fix it according to whatever priorities they have.

[1:00:08]

Anyway, so it turns out to be pretty usable by a variety of people with no training in gender differences, because the materials provide them with what they need to know. And we did a field study in summer 2015, our very first one, and we had two government agencies and two software companies try it out and use it on their own software. And when they evaluated it on their own software without our help – they did it themselves – their average was one out of four features in their own software that they were evaluating, they found gender inclusiveness problems. Usually these were gender inclusiveness problems that disadvantaged females more than males, but there were some that disadvantaged males more than females. So I was just blown away. I did not expect one out of four; that's a huge number, one out of four of every one of their features.

Anyway, so then of course it's up to them to fix, and all of those agencies and companies did something about them. Two of the agencies changed some of their user interface guidelines and standards by going through the GenderMag process. One of the companies immediately fixed three of the things they found. Another one would have immediately fixed some of the things they found, but it turned out those particular things they didn't own. So they had to convince some other group to fix those and that's hard to do, to convince somebody else to fix it when the somebody else wasn't there and didn't see it and they're saying "gender differences, really?" But they stuck with it and did convince that group to fix it. Since that time, a lot of other companies have used it, and the average is actually going up. By this I mean, as more and more people have experienced it, I've started to see data even worse than one out of four. So now the overall average I know about is one in three. One in three features they evaluate in their own software, they find gender inclusiveness issues in.

CP: This is very exciting.

MB: Well, I hope so! I hope they fix them all and we change the world by having software that's not only more gender inclusive but just more inclusive.

CP: Well, if nothing else, you've created a new field of research.

MB: Yeah, I guess so. And a lot of other people are now in it too, so it's not just us. This is all very encouraging because a lot of people is what it takes to make a lot of changes.

[1:03:09]

CP: I have a couple of other themes from your vita that I wanted to ask you about. One because it sounded interesting to me – what is information foraging?

MB: Yeah, so information foraging is this awesome theory that, actually, it has an OSU tie to it as well as stuff that we've done. Let's see here, first I should say it's a theory. So I'm a huge proponent of using theories to guide our computer science research. If we don't use theories, if we just start with intuition and evaluate it empirically later on, we might find out something works or doesn't work later on, but we may not find out what was really important about making it work. It might be some detail that we didn't think was the real reason. But if we also inform our work about our theory, about something that should work and why, then if it does work we can add evidence back to that theory and also have a nice connecting tissue between a lot of different projects investigating that theory. So I'm a huge fan of that; there's a lot of theory in the GenderMag work such as what we drew from theory. And so information foraging theory is another one of these theory-based things that I've been working on.

So it all started with this brilliant guy named Peter Pirolli – actually it started before that, but I'll start with Peter. He was a grad student at Carnegie Mellon in the late '80s and he started thinking about how people seek information when they are on the web. And so he had run across some foraging theories without the word "information," and so this was some work that some biologists had done back in the '60s, including a team at OSU. And so they had developed this theory to figure out how animals that are hungry pursue prey when they are in the wild, like the jungle. And so they put together these mathematical and computational models and had discovered that this theory was working pretty well at predicting, like, where the lion was going to go when he's hungry in the wild and there's some rabbits over there.

So they figured it all out, and then Peter said "well, I wonder if this same theory would apply to people instead of animals pursuing information instead of food on the web instead of in the jungle?" And lo and behold he found that the same models and concepts actually worked in that situation. And so that's how information foraging theory was born. Anyway, that was Peter's Ph D. thesis, and he mostly validated it in document collections, especially in web foraging.

Anyway, at some point much later, in the early 2000s I guess, one of my students Joey Lawrance said "well, if that's the way people pursue information on the web, I wonder if the same thing would also apply to software developers who are trying to pursue the information they need to fix a bug in software itself." And so he started pursuing how information foraging theory might apply to that. And so at this point I guess I should probably – well no, I won't go into the details of it. Anyway, he had very good success with this and found out that the same model Peter and his colleagues had used also worked in software environments – like software development environments, debugging tools, that sort of thing – to predict where programmers would go next in this huge body of source code when they are trying to fix a bug. And then later on, one of my later students David Piorkowski picked it up and started investigating it in other software development situations as well. Still debugging, but more different environments, and he found some very interesting things. And now I have yet another Ph D. student, Sruti [struggles with last name]

[1:07:50]

CP: OK, we just fact checked the name of one of your students.

MB: Yes [laughs] one of my students, Sruti Srinivasa Ragavan, also started investigating this, when there are multiple versions of the programs lying around, how people navigate through that kind of space where a lot of the information copies are almost the same but not quite, and how it helps people through this kind of situation. So yeah, that's information foraging theory.

[1:08:22]

CP: The last bullet I wanted to ask you about was a collaboration that you engaged in with another person whom I've had the pleasure of interviewing for this project: Tom Dietterich, on machine learning. Can you tell me a bit about working with him?

MB: Yeah, working with Tom was awesome. He wasn't the only one I worked with on this project, I also worked quite a lot with Weng-Keen Wong who's another of our professors in the Computer Science department, and one of our long time colleagues Simone Stumpf who at one time was a post-doc at OSU and went on to become a faculty member at City University, London.

Anyway, so we characterize this work as end-user debugging of machine learned programs. And the lead Ph D. student on this was Todd Kulesza, who's now at Microsoft. So anyways, the idea is, when machine learned researchers are building machine-learning algorithms, what those algorithms actually produce is really other algorithms that continue to change and morph depending on what new data comes their way. So the original researchers, what they produced isn't actually what people end up using when they're using a smart application; they're using something that sort of morphed out of it. To give a concrete example, supposing a machine learning researcher like Tom Dietterich or Weng-Keen Wong wrote an algorithm to enable you and your phone to have a restaurant recommender right on it. So they write the algorithm, and then they train it on a bunch of data that they hope is relevant to your situation, and then you download it onto your phone. And so at that very moment what you have is not their original algorithm, what you have is their algorithm after it went through training. And then from the very day you start using it and putting in your own preferences, its morphing again. So now its individual to you. So if it starts misbehaving, you can't call Tom or Weng-Keen because it's not the same program they wrote, so there's nobody around to debug it except you.

Now let's suppose that it's something that's important. Restaurants recommendations, those aren't so very important. If it starts recommending places that have food poisoning and Health Department problems and all that kind of stuff, you can just throw it away and stop using it; it's not going to be the end of the world for you. But there are a lot of smart applications that are individual to you that do matter quite a lot. So let's say for example these aging in place situations. So supposing you have an elderly grandmother who you'd like to watch out for, and suppose she lives not here in Corvallis but in Portland in her own home where she's been forever and ever and ever, and you've outfitted it with smart phone stuff so that if anything bad happens you'll be notified. So of course somebody like Tom and Weng-Keen would have written the original algorithm and trained it on grandmothers the best they can, but then it's in your grandma's home and now it's being trained on stuff particular to her. Well, if something goes wrong with that, this is going to be important, because if it notifies you that grandma's fine when in fact she's lying on the floor with a serious head injury, this is very bad. You need to know so that you can go up there and help her notify somebody or whatever. And so this would be a false negative; in other words, it's just not supporting something that it should be.

You also don't want false positives, so every time a twig blows against a window you don't want to be told that there's a burglar trying to break in, because you can't afford to jump in the car and go up to Portland every time that happens. So false positives are bad too. And there's nobody to debug it but you. And you don't have any debugging training and you can't see the source code, and even if you could see the source code you wouldn't want to. So what are you going to do?

So this is the topic that we were all working on together. And so what Todd did under all of our direction was come up with explanations that could not only explain the logic to the end-user in plain English – explaining why it was recommending things – but be thorough enough that if you explain back to the machine what was wrong with that logic that it would be able to use that and correct that logic. And so we think of it as debuggable explanations.

And so this is really hard, because if you have dumbed down the explanation too much, then whatever it is the machine is telling you is the explanation is basically a lie. And if you try to correct that lie, you're not really going to get anywhere because now you're working on false information that it told you, and furthermore you probably don't have enough detail to really be able to fix what's really wrong. So the system has to be pretty complete, but it still has to be understandable by you and actionable by you. And then the back end needs to be able to take the corrections you made to that explanation and do something fruitful with them, so it really and truly makes things better.

So Todd made a lot of progress in this area and came up with some principles of debuggable explanations – we called it principles of explanatory debugging, is what we called it. Something very exciting happened a little while after he graduated: DARPA, the defense research arm, saw this work and got very excited about explainable artificial intelligence. And they built a whole program informed largely by that work and some other closely related works that other people had done about the same time, so they have this whole research program now around explainable AI, which is one of the things that is kind of the main thing that came out of that collaboration. So yeah, let's hope that soon in the fairly near future, when something goes wrong with a smart application that really matters to you, an ordinary end-user, that you'll have the information you need and the power within the software to be able to fix it so that it starts giving you righter answers than it did before.

[1:15:36]

CP: I have a couple of questions about institutional relationships, and the first is with Hewlett-Packard. Hewlett-Packard has had an up and down history in this community and I was wondering if you were engaged with the campus here in Corvallis.

MB: Not at the moment.

CP: Or if you had been in the past.

MB: Yeah, I have been. And also HP Labs down in California. So back when I got my NYI reward from the National Science Foundation – that was that young investigator award – one of the deals was that they wanted us, the awardees, to collaborate with industry. So if we collaborated with industry, we got a money match from NSF, so it was important for us to do this and to get some support. At that time we had quite a good relationship with HP Labs, so our department chair at the time, Walter Rudd, found some people at HP Labs who were very interested in working together on visual programming languages. So they supported me for two or maybe three years to collaborate with them there. So that was the first collaboration that I had.

And then later, one of our former Ph D. students, Rajeev Pandey, got involved in a research project that they had going on here. He worked for the local campus, and I think he may have had one foot in HP Labs too. There had been some periods of time where HP Labs had a few tendrils up here in Corvallis, and that may have been one of those times. Or maybe Rajeev was strictly on the industry side, but he was still involved in this project. It was called E-Speak. It was a infrastructure thing that they were working on to help support some of the things that eventually turned out to be web services today.

Anyway, so we did a collaboration with him on a new type of visual programming language called FAR which was intended for small businesses to do e-commerce kinds of things without having to go through a professional programmer. So that was the second monetary collaboration we had.

And then we also have some sort of data exchange and just sort of working together, brainstorming things. And the most recent of those, they've been experimenting with GenderMag to see to what extent it might figure into some of the things that they do on the Corvallis campus. So yeah, I think that's everything that I've done with Hewlett-Packard. So kind of off and on, yeah.

CP: The other organization is OSU Space Grant. We've interviewed Jack Higginbotham for this project, and we've gotten into the details and history of Space Grant with him, but I'd be interested in your perspective as the second person that I've talked to about Space Grant.

MB: Yeah, well it's been quite a long time since I was involved with them, but in some of my early years we were doing some visual programming language research that was quite applicable to what they were doing. So as a result of that they supported several of my students with fellowships, but that's the extent of what I did; not a great deal.

Two other organizations that don't seem to be on your list that probably should be – three, I guess – are IBM, Microsoft, and Saturday Academy. IBM – I've been a longtime collaborator with IBM. They supported me with IBM faculty awards two or three times depending on how you count, and I've had a longstanding relationship with some researchers that IBM – T.J. Watson in the New York area, it's a research lab. In fact, a lot of the information foraging theory research was done as a collaboration with IBM. And they too are getting very interested in using GenderMag. In fact, they are just about to release a story about how GenderMag works to an internal IBM design magazine that they publish there, and were talking to some people there too about perhaps using it in a bigger way. So that's the collaboration with IBM.

And then Microsoft has been a supporter on and off for years too; Microsoft Research, usually. And so they were among the first supporters – actually I think it was Microsoft Research, they supported Laura in one of the – Laura Beckwith, one of my first Ph D. student in that area – they supported her to do a study up there so that we could get a bigger variety of populations into our experimental base. And then Microsoft Research in the UK actually supported three years of Laura's Ph D. studies because they were impressed with the research she was doing. And then I did a sabbatical with Microsoft Research in the UK and another one later with Microsoft Research in Redmond, and they gave me a couple of gifts to support the gender HCI research as well. And then this last year, I did a series of visits to work with them on how GenderMag and derivatives of it might work in their organization. So they have a long history of on and off support for that work.

[1:21:14]

CP: And Saturday Academy?

MB: Oh Saturday Academy, yes. They're such an awesome organization. I like to talk about how wonderful they are anytime I get a chance.

One of the ways that I've worked with them is we've had Saturday Academy students as part of our research lab almost every year – I think over the last ten years, something like nine of them. Anyway, so what we do is – I also have quite a number of research undergrads on my team and a lot of grad students and, as you've heard, some graduating professors. And so we just operate as one big team, we don't divide each other up, and so in the summers we bring in two high school researchers through Saturday Academy's apprenticeship in science and engineering program. So that's this awesome program where the top students in the high schools can apply to be in this program. And they need recommendation letters from their teachers and that sort of things, and then potential mentors for these kids across mostly Oregon but I think there's some southern Washington and a little of northern California too. We'd all apply – scientists and engineers in that area – to be mentors. And then there's kind of a matchmaking process where the students who've been accepted and the mentors who've been accepted, the students can apply to some of the mentors programs that look interesting and then the mentors can interview the students that apply to them to sort of rank and order – accept or not accept – into the summer experience. And Saturday Academy does all this matchmaking and makes sure that all the mentors have passed all the right tests to make sure were not felons and that sort of things. And then they help supervise as well, so they give mentoring training, and they provide a high school teacher to sort of make sure everything is going ok. Anyways, so then the students work full time for something like eight weeks over the summer, and that's their way of learning whether that branch of science or engineering is something they enjoy.

Anyway, so I've had wonderful luck with that program; some of the students have gone on to actually major in computer science, some of them here at OSU. A couple of them have actually turned into my research undergrad students. So I've been very, very happy with that. The other way we've worked with Saturday Academy is we've collaborated with them on grants. So sometimes when we do a grant application it has impacts for education as well as for research itself. So we partner up with Saturday Academy because they know a lot about education and work together with them to gather data about how it works for education, and to offer courses through them to give high school students a chance to try these things out. So yeah, we've had wonderful experiences partnering with Saturday Academy and their OSU arm, which is now called STEM Academy. And yeah, it's been a wonderful collaboration, and anybody who ever has a chance to do anything that might have an impact on high school students, you should use Saturday Academy or STEM Academy depending on where you are, because they are awesome.

[1:24:49]

CP: You've been a real champion of undergraduate research, how did that emerge as a focal point for you?

MB: You know, it's kind of interesting, it's something I drifted into. When I was a new professor, there was a new program called Research Experience for Undergrads Supplements. And so it was something that the National Science Foundation offered. And so if you already had a grant from them, you could get these supplements to bring in an undergrad student to work with your team. They particularly encouraged you to bring in members of underrepresented groups in your field, and so for me this would be women in computer science. And so I thought, "ok, I'll try this." And I did and I had a very good experience with it. And I realized this makes a huge difference to an undergrad when they are in it, the same way that one mention of grad school had made a huge difference to me when I was an undergrad.

And besides that they were so good, there was never a situation where I felt like I was sacrificing, I was gaining from the experience as much as they were. So I thought, "this is awesome!" [laughs] And so, at the time I did it, I was the first here at OSU to ever do a research experience in computer science for an undergrad, but I just kept doing it because it was working out so well for everybody. And later on, more and more people through the department and eventually the university started doing it too. So it caught on, and I love it. I love doing it.

CP: I have a question about balancing family life with your work. I came across an interesting quote where you mentioned protecting your evening time from 5:30 on as family time. I might be presuming wrong, but I assume a lot of high-achieving academics have not been able to pull that off. Will you talk a little about that?

MB: Yeah, so it's really just an extension of time management, really. So for me, I always regarded it as building fences around certain portions of the day. And so when my kids were little, it was almost forced that way because the day care closes at 5:30, so you know you have to go pick them up. So I built these fences so that when the day care center closed, that was family time, and then when I dropped everybody off the next morning, that was work time. And I also drew a fence around weekends. So it's just a matter of knowing when you allow yourself to be doing this or that rather than always feeling guilty when you're doing the one situation where you're not doing the other. There were occasional times when I made exceptions; there might be some horrible deadline where I might work on a Saturday. But mostly I stuck by the fences, and I did that pretty much until the kids left home for college. Then I got a lot laxer about it. [laughs]

CP: Well, I've got a few concluding questions for you. The first is if you could just provide your sense of the environment for women in computer science these days in the field.

MB: Well it's better in some ways, but there's just such a long way to go. Some of the stories that were around back when I was a younger woman, well we didn't know about them as much, but some of the stories that date back to those times, those same stories are still around. We still have glass ceilings, we still have the feeling that we're not where we belong and we're not feeling welcome, and that somehow whatever it is we're doing isn't quite right because it isn't the way the men do it. And so all of this is still around in our education environments and our workplace environments and even in the software.

But there's one difference – well, more than one difference. One of the big differences is we know a lot more about it now as a society. And so as individual women, when something inappropriate happens, we now have the knowledge available to us that that's inappropriate, and it's not our problem, it's their problem. And I think that, all by itself, is a real step up towards being sane and getting through this sort of thing. So that's a good thing.

Another good thing is that business has come to recognize that diversity is very important for good business outcomes – for making money, for example. So they now understand the importance of diversity, whereas in the past they didn't. So what this is, is business shares the goals that many of us have for other reasons, which is: diversity is good for everyone. It's good for creative thought, it's good for making money, it's good for understanding a larger portion of our customer base, and so that's another very big difference. Still having the goal of diversity and actually doing the things required to get diversity, there's still quite a big gap that we see in a lot of companies today about that. So, you know, there's been a lot of press about Silicon Valley, and they're all releasing their numbers and guess what? The numbers aren't very much. It's like, we keep saying it's important but we're not making any progress! A lot of the times they aren't necessarily drawing upon best practices and really getting it into the reward system well enough. And so it takes more than lip service. You know, it's hard work to get there, but as business knows, getting there is worth it. It's worth it for the business and it's worth it for society.

[1:31:26]

CP: Can you talk a bit on – you've been here for quite a while at this point – the changes that you've observed within Computer Science and within the College of Engineering over the time you've been at OSU?

MB: Well, there are a lot more women in the College of Engineering at the faculty level. At the student level, there are more women as well; not as many more as we'd like but still more. So that's been very good progress. In the past decade, one change is that we now have a Director of Women and Minorities in Engineering at the college; we didn't have that before. That's not a new, new change, that's like a ten-year-old change. I think the importance of diversity has made it all the way up to the top levels, so that's something that's been true for maybe the last seven or eight years, not the last twenty-five. So that's another big change at the university level. In Computer Science, if what we're talking about is diversity –is that what we are talking about still?

CP: Well, broad change. I think that from the outside, looking at the College of Engineering, it's been a point of emphasis for this university in the last couple of decades. New buildings, lots of fundraising, and I have to imagine it must be interesting to view that from the inside.

MB: Well, I mean computer science is – it permeates everything. Furthermore, the state of Oregon, we're no longer a timber state and a fishing state and an agricultural state. The driving economic force now in the state of Oregon is technology. So of course it's very, very important to the College of Engineering and to the university to be offering the education, the training, that our people need to be able to thrive in that kind of economy. And so I think that's one reason why Computer Science and Electrical Engineering, the whole school of EECS, is so large in terms of students.

We still need more faculty to teach them. For example, just this year, the junior level course that I'm teaching this term is fifty percent larger than it's ever been in any of the past terms I've taught it. So that's a huge increase. It's a challenge. There's a huge bubble coming in right now, our enrollments are skyrocketing, and that's why it's just critically important to the state of Oregon.

CP: Well, the last question is one we've been asking many people – just to lend their thoughts on the university as it looks towards its 150th birthday. Where do you see OSU as being positioned now and where is it going?

MB: That's an interesting question. I don't know that I can speak for all of the university, but I'll sort of confine my thoughts to Computer Science because that's the part I know the best. I think that we are at the forefront of computer science of a lot of changes. One of them is the importance of mixing our areas. So way back when I was getting my degrees, there were these well-defined labels, and you were one or another but not a mix. And nowadays people are kind of a mix. And so we have, for example, bioinformatics faculty members.

And we have – just about every project that most of us faculty are involved in nowadays are collaborations. In the old days, one professor and their students did one project, or maybe two of them, but now most of what we do is as part of the group, and in that group there are people with multiple specialties. And so there might be an HCI person and a machine learning person and a software engineering person all working together on something. So that's been a huge change for us over the years and it's, I think, the way of the future.

I think in a way it comes back to diversity of thought, and here there is diversity of specialty and talent as well. The more perspectives you can bring to a problem, the more likely that you can get out of whatever boxes it's been in before and come up with something creative and game-changing in any area. So yeah, computer science has never run out of steam, and the whole time I've been working in it there's always something new and more interesting than last year to work on. It's a wonderful profession to be in and an opportunity for life-long growth. And yeah, I feel like I really got lucky; I got the best job that there is for me.

CP: Well, I want to thank you for this. This has been very valuable for our project and I wish you the best of luck as you continue.

[1:37:03]