Oral History of Margaret Hamilton

Interviewed by:

David C. Brock

Recorded April 13, 2017

Boston, MA

CHM Reference number: X8164.2017

David C. Brock: Thank you, again, for joining us here this afternoon. I thought we could begin at the beginning, as we discussed, and talk a little bit about when and where you were born.

Margaret Hamilton: I was born in Paoli, Indiana. And I'm not telling what year. <laughs> But it was 1936 in August.

Brock: And could you describe where you spent your youth and what sort of community it was?

Hamilton: Okay. I left Paoli when I was six weeks old. So I don't remember too much about that part of it. But I do know that my family moved from Paoli, Indiana to other parts of Indiana, for example, Bloomingdale. We also moved to parts of Michigan, Ohio. Midwest, basically was where I spent a great part of my life. And I don't know how far you mean, when I was young, but I guess in high school is where I ended up in Michigan, in the upper peninsula of Michigan. So I was used to being in classrooms with people I didn't know, going to new places and living in different places.

Brock: Was that moving around through high school age was that because of the activities of your parents, I assume?

Hamilton: Well, when I was born it was before World War II. So some of it had to do with the war. And some of it had to do with finding different-- teaching in different places. But both parents were involved in teaching. So it was-- or it was living with grandparents' part of the time.

Brock: And could we talk a little bit about your parents? You mentioned they were teachers. Just about them and about their kind of most notable characteristics.

Hamilton: Yes. Well, my mother was a teacher in high school. So she taught most of the time that I remember. It's hard to remember that many years ago because-- I think most of the time she was teaching. My father was a professor in college. And he taught English, philosophy and poetry. And he was in the war during World War II so he took a break from teaching. And he was also president of the Michigan Poetry Society. So he was quite active in the poetry end of things; his favorite pastime.

Brock: And had they both been in Indiana for a long time by the time you were born? Or was that…

Hamilton: Actually, they had met each other in the Upper Peninsula of Michigan in a different place than where I went to high school. Near Escanaba, Michigan. And it was a little town called Garden. It would be lucky if I could say it had 100 people. It was tiny. Which is like in Michigan, I lived in a town of 200 people. So it was a lot of little towns that I found myself in for one reason or another.

Brock: So would you describe the context in which you grew up as primarily rural?

Hamilton: Well, I've lived in places where there's woods that might fall under that. But they were usually, as I remember, little towns with the main street that was tiny so you knew all of your neighbors. So it was a town as opposed to being out in the country normally.

Brock: Well, if you had to describe it, what would have been sort of the primary interest of your parents' household? Was it the arts and poetry, learning?

Hamilton: Okay. Well, first of all I have two separate families. One is my father's side. One is my mother's side. I guess everybody was into reading and the arts. In fact, all the time growing up there was piano on both sides, music. Always music. So both grandparents were ministers. The Quaker, on the Quaker side, my grandfather was a Quaker minister. And on the other side grandfather was a minister in the Protestant churches. And being in a small town he'd go from town to town to be a minister to different areas. And my grandmother played the organ in all of these different churches. On the other side, on the Quaker side, everybody played the piano. Everybody was good at it. I was not that good at it but everybody else was good at it, my grandmother, her daughters. And it was always maybe three pianos being played at the same time by people in the family. And yeah, on the Quaker side of the family it was my grandparents spoke "thee," "thou" and "thy," so it was a whole different language there than it was elsewhere. So we were influenced by this. And, in fact, I think I may have mentioned but one of my favorite books was "Thee, Hannah**"** and it was about a little Quaker girl. So that was one of the things. But we always had books around us and music.

Brock: And you mentioned that you didn't consider yourself to be as good a pianist as other people in your family. But did you enjoy music? Did you enjoy reading?

Hamilton: Oh, I loved music of all kinds. But most of the background was classical music. But as I was growing up, I started liking things like Elvis Presley and The Doors. I was sort of rebelling against that[1] but still I was always around music and I always loved music. I just was not a great-- my sister is a concert pianist. She's younger and she put me to shame so I decided this was not my thing.

Brock: Was it in the Upper Peninsula where you spent your high school years?

Hamilton: Yes.

Brock: Could you tell us a little bit about what those years were like for you?

Hamilton: Okay. I'm not sure, which angle to answer that at.

Brock: Maybe your life inside the classroom. Was school easy for you?

Hamilton: Oh school, I loved all courses and I really liked school because I liked my friends. I could see my friends. And I remember always liking the teachers I had. But school for me was fun. I looked forward to it. And I loved every course-- I shouldn't say loved. That's a strong word. I mean math was my favorite. And what I did not like was home ec. [home economics] because girls were supposed to take it. And, again, it was rebelling against what we had to do. So that was my least favorite.

Brock: And did you have any particular hobbies or extracurricular pursuits?

Hamilton: You name the hobby, I had that hobby. I did-- I mean you could just say anything, "Did you do this?" And I'll say "Yes." I was in a choir. I played in a band. I went horseback riding. I liked dancing. In fact, I might think of a favorite one I haven't mentioned at this point. But yes, I really liked doing things. I loved playing baseball. It was an all-boys' team but we were in such a small town that they needed an extra player. So they asked me if I would do it and I was thrilled.

1 Hamilton adds August 2021: That is, rebelling against classical music.

Brock: What position did you play? Do you remember?

Hamilton: I don't remember. I was not the best player on the team, but it was fun.

Brock: Could you speak a little bit to mathematics in your high school years or before? And to the extent to which you can recall what was it that-- was there anything in particular that attracted you to mathematics.

Hamilton: I guess I liked the theoretical side of things the most, abstract mathematics. I didn't really like numbers as a thing. In other words, I was not really-- I didn't like things like statistics. Applied math was not my favorite part. I mean I was good at it, but I liked the kind of things where you derived things. And I liked geometry, algebra. This is now in high school. Right? But yes, that was the side of things.

[Crew Talk: Brock and Hamilton asked to repeat last question.]

Brock: So if you wouldn't mind talking about the sort of mathematics that appealed to you versus that which didn't so much.

Hamilton: Right. I liked the abstract side of mathematics which was always the case. So instead of memorizing things, I liked to derive things. Even when I had an exam, people would memorize them. I would always feel safe because I could derive it to get the answer. And also, I liked things like algebra, geometry, logic. I was not that fond of the applied side as I was of the other side. And I didn't like things like statistics as much as I liked the other side. So I always stayed sort of abstractly thinking about things. That was-- and it's always been the case.

Brock: Were you at all attracted to technology or science in these years?

Hamilton: I was good at it but keep in mind we were in a very small town. So whatever they taught I would take. I mean I remember in biology I didn't like seeing the heart of a frog on a board. That was upsetting but hey, it was in the course. But yes, I took physics. Yeah, the things that they taught we took. But I was not-- nobody pushed me in a certain way. There were just a few courses and you took what they had.

Brock: As you were approaching the end of high school what were your-- how did your thinking for yourself develop about what you might do next? And what you might do in life as a young person at that time?

Hamilton: Well, I definitely wanted to go to college because that's what you did in my family, both sides of the family. And so it was just what you did. You didn't really have to think about it. Although, my father, I did tell him in my rebellious way I decided not to go to college. He said, "Fine, that's okay." Well, I quickly changed my mind. <laughs> So that's yes. And what did I want to be? I really wasn't sure. I mean my grandmother on one side was a journalist. And my grandmother on the other side was a musician. Because of my father-- my mother was a teacher, as I said. And my father was a philosopher. So I always thought I would like to get into something that was interesting. But I was also wanted to keep in mind that I had to make a living. So I had the two parts to weigh.

Brock: And do you recall where you explored going to college?

Hamilton: Well, yes, I do some of the things. And by the way, a lot of things that influenced me too is that I had worked on many jobs in high school and then going on to college. So that gave me an idea of the kinds of things that I like doing. So I had a little help from that.

Brock: Would you mind telling us a little bit about those work experiences?

Hamilton: Well, I was always working because I wanted to earn money to go to the college that I wanted to go to. And we were not a wealthy family. There were four kids. And so if we wanted something we had to earn it. So I earned everything possible that I could but saved up enough to go to college. But I worked on many different jobs. I waited on tables in the resort area. But the most interesting job of all was when I worked at the Arcadian Copper Mine. And when I worked there it had been a copper mine, abandoned, turned into a banana mine because it was 40 degrees and so they could store bananas in there. And this guy got this bright idea of turning it into a tourist—to go on tours. So at the time he hired me as a guide to take people on tours through the copper mine. It was native copper. So anyway, I started taking people on tours and they started off with like a family a day, two or three people a day. And then pretty soon it started growing. And I had to start hiring other guides to work for me. And by the end of that year it was like at least 1,000 people a day over the summer. And so I had these guys working for me and one time they went on strike because they weren't making enough money being guides. And so the person who owned it – his name is Arvo Walitalo – and I said, "Arvo, these guys are going to leave unless you get them a raise. And by the way, I make the same thing that they do. So could you give me a raise too?" And so I managed to get a raise for everybody. And then the people were coming in. And then there was-- I don't know whose idea it was but to sell copper jewelry, copper gifts. Right? So pretty soon I took over responsibility for that. I hired my brother as the head guide. And then hired people in the jewelry department. But since Arvo never had education above fourth grade, and he owned like three Cadillacs and he had money, he also let me handle the financial part of the business. And so over this several years I was running the Arcadian Copper Mine to take people on tours for quite a long time on summers during both high school and college. I also worked at Joe's Chicken Basket at nighttime when I was running the copper mine. So you can see I got an idea of what you're getting into when you're going to work every day and for responsibility and what it means to have responsibility. And all of this started when I was sixteen. The copper mine. And so that had major influence, I think.[2]

Brock: It strikes me that that's a lot of trust that the owner of the mine with the fantastic name, that's an unusual amount of trust to place in any young person.

Hamilton: Well, remember he never went past fourth grade. And he thought that what I did was all he needed, and I felt the same way. But remember there's only like a family starting off a day, hundreds of people a day as it grew. And it was one of the most popular tourist attractions. And so I remember one time when I had found a huge piece of native copper. And this is the only place in the world where they have native copper. It's pure. Some have threads with silver or whatever. Whereas other parts of the world have copper but it's not native. They have to crush it to get the copper. See I still have something left from taking them on tours. Right? But I remember finding this huge piece of native copper. They had never seen anything that large before in the mine.[3] And so I decided that if you find it it's keepers. Right? So I show Arvo what

2 Hamilton adds June 2021: I worked at Woolworths before this starting when I was fourteen.

3 Hamilton adds June 2021: We were told by Arvo that we could keep any loose copper we found in the mine.

I found and he said, "No, no, that's too big. Nobody gets to take one that big." And I got really upset. And he said, "Okay, let me think about it." And he came back and he solved the problem. He did a Solomon solution. He cut it in half. <laughs> So I still have half of that native copper. But yes, my life was very much in high school and college summers doing that.

Brock: Well, that's incredible.

[Crew Talk]

Hamilton: Also I worked in college during the winter time too but I don't know if you want me to go…

Brock: Sure. We'll continue with your work experiences.

Hamilton: In college, I worked in what they called the scrape line: cafeteria. Okay. So we used to have all kinds of names for that like, train wreck or whatever. This is college kids. Right? So I worked there but I also, again, worked at nighttime on switchboard because I was like the phone operator on switchboard. And so it was a place where a lot of people would come at night to pour out their problems, because there I was, telling me about this or that. So I had a lot of friends. But I remember one time there was a water department, but there were two water departments in the area. And so they would call and they would hear each other because I would plug them into each other. And they'd say, "Hello, this is such-and-such water department." The other one says, "No I'm the water department." So I had fun with the switchboard as did the people who would come to visit. But I did that in the evenings and the scrape line. I actually did it during the daytimes too. And the scrape line whenever there was a meal. I can't remember. But both of them kept me busy in addition to studies. So, again, I was used to working.

Brock: And was that with the Bell System, the switch exchange? Or was that on your campus?[4]

Hamilton: It was a thing where you plug stuff in.

Brock: Right. Yeah. Okay.

Hamilton: You know, like you see these things with that comedian, you know[5].

Brock: Of course, yes.

Hamilton: So it was familiar to me. Yes.

Brock: How did you decide where you wanted to go to college?

Hamilton: Well, first I went to the University of Michigan. And probably because I had a good scholarship there. And yeah, I think, I had a good scholarship is the main reason. I don't remember why I started there over something else. I had scholarships, complete scholarships to other places like Stevens College, but it was all women. And I said to my father, "I don't have anything against women, but it's much more real world like to have everybody in the college, not just say because you're a girl, you're going here or you're a boy you're going here." So I decided that was not a good idea. It should be what it is like in the working world, a realistic situation. So then after the University of Michigan, if you recall, I came from very small towns, and it was a little big and not as personal. I did well in my courses and had many great experiences. And there

4 Hamilton adds June 2021: It was on our campus.

5 Hamilton adds June 2021: Lily Tomlin's telephone operator character Ernestine.

were things I did that decided what I might do next. But I eventually decided, you know, after I'd been there for a semester to go to Earlham [Earlham College] and try out a smaller school. Keeping in mind, I was so used to moving around, it was no big deal to try this out and see what it was like. But I remember when I was at the University of Michigan I studied really hard because it was this big school and I came out of this little high school. And so I was really worried and I stayed up many, many nights for the mathematics exam. And all of a sudden it hit me I hadn't studied for sociology. And it was like the night before and I was-- being up all of those nights I wasn't thinking clearly. And I thought, I've got to do something. So I remember that night before the sociology exam I memorized the entire glossary. And it was a blue book exam which said, "What is man?" And I wrote this entire thing for whatever two hours or so, weaving in every term I had memorized. And the sociology department called me in and said, "We think you should consider majoring in sociology." <laughs> So anyway that was memorable. But nevertheless I decided to-- and Earlham had been a family school. My mother and my aunts went there. My grandmother, my grandfather went there. Yeah. And it was Quaker. So I thought I would try it out.

Brock: Had you been to the campus?

Hamilton: It wasn't Quaker. Quakers started it. Sorry about that. Had I been to the campus?

Brock: Before?

Hamilton: I don't think so but I heard so many stories. My uncle went there I mean going way back into the 1890-something. Right? And I'd hear stories about it. Oh my goodness, so many members of my family went there. Yeah.

Brock: Well, before you-- just about mathematics at Michigan, how did you find that? Were you encouraged by the mathematics experience that you had at the University of Michigan?

Hamilton: I did well. I mean I think I got all As far as I recall but don't check with the college. <laughs> I might have a bad memory. But anyway, I did do well at it. And I was encouraged to pursue mathematics. And yeah, and I took the courses. I don't remember which courses but I took more than one. And that was my major, let's put it like that.

Brock: Right.

Hamilton: And was I encouraged? Nobody encouraged me to do anything from the family side of things, from the teachers. There were just these courses and you said I'll take that one and that one and nobody questioned you one way or the other, encouraged or discouraged.

Brock: And I know that at Earlham or I have read that you also combined studies of philosophy with studies of mathematics.

Hamilton: Yes.

Brock: Did you begin that at Michigan, as well, do you recall?

Hamilton: No, I didn't do it at Michigan but it may not have been a course that was in the particular time. But I did want-- I really was fascinated-- actually I may have thought about it because it's not clear but I may have at that time majored in math and had a minor in philosophy and religion because come to think of it I think we visited in a religion course every kind of religion in the area in Ann Arbor. And I was fascinated by the commonalities between all of the

religions and then the differences. But I didn't think I'd make a very good living studying religion. So I had to balance it anyway. But somehow my father and I were convinced that philosophy and mathematics were definitely connected. So if I was an expert in either one it would help the other, at least the kind I liked.

Brock: Before we get into your kind of how your education and mathematics continued and philosophy continued at Earlham I just wanted to ask about computing and computers. Had they entered into your life in any way by this time?

Hamilton: Not in high school. We never heard of these things. In college, not until I think it was 1956 or 1957. I didn't hear about from college, but I worked another summer job. I worked for Travelers Insurance as a student actuary because they paid you to study. Okay. And so you had to work part of the time but then you took classes part of the time. So they put me into this room full of Marchant machines; like, mostly women. In fact, it may have been all women, in there, number crunching. They were like computers doing all of the statistical stuff which was not my favorite kind of thing to be doing. And there were several insurance companies in Hartford [Connecticut]. They'd all get out of-- each would get out of the different time because there was so much traffic. So let's say that Travelers got out at 4:33. I mean it was that odd. So everybody at just around 4:30 everybody in the whole place would watch the clock. And then it would go to thirty-three and they'd rush out practically stampeding. I remember sitting back there saying, "Oh my God, I don't want to do this for the rest of my life." <laughs> I couldn't believe how anxious they were, which I knew anyway, that I didn't want to do it. But it was just an experience that I remember. And then part of that experience was being called in by the VPs and the EVPs wanting me to major in becoming an actuary. And when I found what was involved and the fact I did not like statistics but they really tried to get me to do that and I decided, no I didn't want to do that. But I had a great teacher at Earlham. Her name was Florence Long. She was the head of the math department. And it was having her, I think, for the teacher, that, I mean she was so good at it. But she was also a wonderful human being. She would invite all of her students, myself being the only woman and all of the guys, over to her house for cucumber sandwiches. I mean it was like a family kind of thing. But she was good. And I remember thinking, "I want to do what she's doing. I want to teach the kind of mathematics she's teaching. And I want to teach it in college like she's doing." So that, major influence that was.

Brock: And she was-- did you talk to her about that ambition? Or could she see it?

Hamilton: I don't know that I talked to her about it. But she really liked what I did in math class. And, of course, kids like to take it when people like what they're doing, even if they didn't. Right? <laughs> It was positive reinforcement. So, again, nobody said-- well, except for the actuary people and the sociology people. But with respect to math, if you did well you knew you were doing well. But I wasn't like these people that I'm driven just in the math direction, where I have a nephew is like you name it and he's a math whiz in everything math--but nothing else. It was not something that was meant to be in my head.

Brock: If I could ask you just one follow up question about that experience at the insurance company in Hartford.

Hamilton: Oh, I forgot to tell you. Yeah.

Brock: Well, just to finish that thought, but that room filled with all of the desktop calculating machines. Was this just a gigantic open room with the loud clack of these--

Hamilton: Like a big auditorium. And, in fact, I remember when it clicked on 4:33 somebody would be in the middle of a calculation on the Marchant machine and they wouldn't even finish the number that they were putting in. <laughs> They just went running out. That's how anxious they were to get out of there. But I forgot to tell you why I brought this up. I brought it up because all of a sudden one day word was going around Travelers that there were these new things out there called computers that were going to take away all of their jobs, all of the people on the Marchant machines. Pretty soon they wouldn't have jobs. And so everybody was talking about it. They were scared they wouldn't have a way to make a living. But, of course, it ended up being more jobs were created with the computers than there were with Marchant machines.

Brock: But you had just heard about the computers.

Hamilton: Yeah, nobody talked about it. No. It was not where I went to school. Yeah, no, because they didn't have courses that concentrated on what we call software engineering or our brand of computer science at the time. It was mathematics or physics but more often mathematics that was what we did. So eventually that became important when they were looking for people in these fields when they became fields.

Brock: As you reached the end of your time at Earlham and this picture of becoming a mathematics professor like Florence Long, what were you thinking about as your next steps?

Hamilton: Well, I wanted to go to graduate school. And I received assistantships, scholarships in various places. But I had heard there was good theoretical, abstract, pure, whatever they used at the time, mathematics at Brandeis University. But first I had gotten married in the meantime in '58. And my husband at the time had one more year to go. So I taught mathematics in junior high and high school in Boston, Indiana, another tiny place. And I remember I had had pneumonia, my senior year. And I was in the hospital for three months with pneumonia. And I couldn't interview for jobs because I was too sick. <laughs> You can hear it had its influence. So the principal and maybe he was the head-- I don't know who he was but they came to the hospital to do an interview for me to be a teacher, a mathematics teacher. And when they found out I studied French for one year they asked me if I would also teach French. And being young and not knowing what I was getting myself into I said, sure, why not? <laughs> So I taught mostly mathematics courses, but I did teach French in one course in Boston High School, while my husband was finishing his last year at Earlham.

Brock: And when he finished then you--

Hamilton: When he finished, then he wanted to get a Ph.D. in chemistry. I wanted to get my Ph.D. in math. We both had scholarships. So then we had to decide-- and let's see in 1959, at the end of '59 I had my daughter. Okay. So now we had a family to support. One of us went to school first, and the other one afterwards. So I don't know how we decided but I probably said, "You do it." And he said, "No, you do it." It was that kind of thing. We both had intended that whoever was working would support the family. And then the other one would go to school and then do the opposite. Now, when he was at Brandeis one of his friends was at Harvard Law School. So he went to visit him in class because he was intrigued by politics and law. So he went there and said, "I think I would rather be a lawyer than be a chemist." And he said, "And I'm going to Harvard." So he went to Harvard for a job and they hired him-- not hired. What's the

word, admitted? That doesn't sound right. <laughs> But he was accepted. And so then we went-- and then I was getting work so I could earn the money and delay Brandeis until it was-- until he had finished his law school. So I can't remember. I think that might have shortened his schooling because of the law school. It might have been. I don't remember. But it was maybe a little less than graduate school and doing chemistry.

Brock: Well, what sorts of work did you initially find?

Hamilton: My first job was at MIT. I worked for Professor Edward N. Lorenz, professor in meteorology.[6] And he wanted to hire somebody to do programing for him. And I don't know how that happened with me. We found each other, let's put it like this. But I don't remember how that happened. And so I went to work for him. One of the best things I ever did. And I hadn't really been near a computer before. He had an LGP-30 in his office. And he had at least two Ph.D.'s if not three. Neither-- none of them I should say, was in the field of computers and programming--and software--, but he loved that computer. And he taught me everything he knew about the LGP-30-- well, he'd hand me something saying, "Here's the instructions." That's how people used to do, right. And but he would show me the things that, like, for example, the LGP-30, it had a drum on it. And if you kept writing instructions without skipping certain parts in the drum, it was slower. So he showed me how to find the places to move the code to go to where it was faster. And he was amazing. He was the most intelligent and one of the nicest human beings. I went from Florence Long to Professor Lorenz. I was lucky. And there were some times there I really remember I would-- I would worry-- I would worry about things. If they didn't work, I didn't want him to know <laughs> I made a mistake. So at 4:00 in the morning, we were all at a cocktail party-- not Professor Lorenz-- I and my friends, and all of a sudden I said, "Oh, my God. I forgot to fix something." I went in at 4:00 in the morning to his office, fixed it. The next morning he said, "What were you doing here at 4:00 <laughs> in the morning?" He calculated. He went back on the equations and figured out where I was and what I had done. And also, I was very impatient because it took so long and the computer, if you made a change, then you'd have to wait for the paper tape to come out, a new paper tape. I don't remember the details so much, but I figured out that instead of creating a new paper tape when I made a change, if I punched a hole in it, because it was in binary-- I was programming in hexadecimal-but the paper tape was in binary. But I poked a hole in there. That was the way to get a 1 when it was a 0. And if I put Scotch Tape over it, I could go the opposite. So I used to take the paper tape all the way down the hall in the meteorology department and make my changes. And he thought that was so funny. <laughs> But it made things go faster. So that was my discovery. <laughs>

Brock: So it was faster to cover up the holes with bits of tape.

Hamilton: To cover up the holes if it was a zero-- I can't remember which way <laughs> was which.

Brock: Right.

Hamilton: Or to poke it with a pencil to create the opposite. So I could change from 1 to 0 and so I knew binary very well <laughs> at the end of that and I knew hexadecimal quite well as well.

Brock: What sorts of programs were you writing?

---

6 Hamilton adds June 2021: The father of Chaos Theory.

Hamilton: It was weather prediction programs.[7] And the only people writing it were myself and Professor Lorenz.[8] And I learned another thing. That the larger the matrices, the eigenvectors or whatever, the longer it would take, and it took a long time in that computer. So I would select the largest matrices to multiply before lunch.

<laughter>

Hamilton: And so, yes. And later on, years later, like maybe five years ago, I realized in one of his well-known papers, he thanked me for all the programming.[9] I never knew he did that. That really meant the world to me, even to this day, that's how much I thought of him. But he was, oh, my God. He was very shy. So there'd be a hurricane, a big one. He'd call me over and say, "Look." And we'd watch the hurricanes. But at coffee time, if the cream was souring, he would say, "Oh, look at the convection currents," and he would look at his <laughs> curdled cream in the coffee. He was a real gem.

Brock: And I would imagine with these weather simulations that it would have to be sort of a collaboration to put whatever he was thinking about for the model to get that, you know, with your collaborating with you into the computer. Is that true?

Hamilton: Yes. Yes. But I don't remember too much about it except he was showing me how to do things. Even, you know, we didn't have operating systems on that. But he showed me what to do to get around it. It's called an operating system <laughs> today. But I mean, a very, you know, small core size and knowing the difference between the sync and the async kind of thing. But I learned so much from him. And again, his enthusiasm, as I like to say to people, was contagious. I think I decided after that I wanted to go into this stuff, but there was no field. But I really wanted to do the kinds of things that I learned from him. And also, I worked over at Project MAC, as part of working for him, on the PDP-1.

Brock: Could you describe that Project MAC scene?

Hamilton: That was a fascinating scene. I used to take my daughter there as a baby. She'd sleep. We'd work-- I'd work at nighttime. The hackers were all there. So we all knew each other. And one time I went in and my program was not working. I was really upset. And when you're kids, when you're beginners, you're never supposed to blame the computer because that's what beginners did when they were programming. And I said, "I know it's the computer. I just know it's the hardware. It's not my software." Right. So anyway, it turned out it was the hardware and the hackers had done something to the hardware in the middle of the night and they changed it, not to make it not work, but I don't know for what reason. So now, they now credit me with catching them in the act and they had to-- they weren't allowed to do whatever it was they did anymore. So there's a chapter that they talk about <laughs> my coming in there, you know, ruining their fun.[10]

<laughter>

---

7 Hamilton adds June 2021: Programs for his original work on Chaos Theory: https://www.wired.com/story/these-hidden-women-helped-invent-chaos-theory/

8 Hamilton adds August 2021: Later I hired and trained another person to write software for Lorenz.

9 Hamilton adds June 2021: https://www.wired.com/story/these-hidden-women-helped-invent-chaos-theory/

10 Hamilton adds June 2021: See Steven Levy's *Hackers*, Chapter 5: "Midnight Computer Wiring Society"

Hamilton: So anyway, that was while I was working for Lorenz as well.

Brock: What was your role on that Project MAC?

Hamilton: It was still weather prediction.[11]

Brock: Oh. To put that on the PDP-1?

Hamilton: Yeah. On the PDP-1.

Brock: And what was the different-- what was it like working on that computer versus the LGP-30?

Hamilton: LGP-30. You know, as I would learn about computers the big thing back then is what language do you know? Okay. Okay, what computer are you on? And after I'd been on two or three computers, I said, "You've seen one language, you've seen them all." Meaning I could just see what's the difference. Yeah, you had to learn the syntax but not the semantics, right. So I never really viewed it in my mind. It became patterns or, but it wasn't like the PDP-1 assembly language versus the LGP-30, yeah-- what is the-- whatever it might be.

Brock: Machine language?

Hamilton: Binary or hexadecimal or whatever it was called that was on that computer. I can't remember if it had a special name for some language that evolved. But yeah. So now when I say, "What's the difference," I would say, "What's the same?" <laughs>

Brock: Uh-huh.

Hamilton: You know?

Brock: Right.

Hamilton: But the environment with the hackers was fascinating, because they would keep bringing in pizza and, you know, they-- And they never-- They were all in MIT in class, but they all cut classes. Right. But we'd be there and my daughter would be sleeping away and I can't remember when this happened, but at one point, there was some program where they sat her down. I don't know if it's the hackers or the group over at the 704, and they sat her down and had it talk to my daughter, say things. And it said she was bad. It was-- I think it's destroyed her to this day <laughs> because she said, "I'm not bad," you know. But it was like a-- it didn't talk to her but the kids-- and I called them kids because I was at least two years older.

Brock: <laughs>

Hamilton: The other kids would read to her what the computer would say. Yeah. <laughs>

Brock: And to tease her in essence or, yeah.

Hamilton: Well, it was more like it was an experiment, a man-machine interface. This was a girl-machine interface. <laughs> As to how it could relate to people, yeah.

Brock: And where-- was that Project MAC group, was that largely men? Were you-- were there other--?

11 Hamilton adds June 2021: Again, building software for his original work on Chaos Theory.

Hamilton: It was all men.

Brock: All men.

Hamilton: And me.

Brock: Huh.

Hamilton: But I was not on their project. I mean, well, their project was hacking, right.

<laughter>

Hamilton: But they thought of me as the establishment because I was not a hacker. However, they considered me an exception because the way they talked about girls, they didn't put me in that category. I was a programmer.

Brock: Ah.

Hamilton: I was a serious programmer. I was there to build real systems, you know. But I wasn't-- I mean, the attitude towards women back then, that was the epitome of it with the hackers. <laughs> I mean, girls were to go out with, right. But I don't think that they ever worked with them at that time, and that wasn't part of the world.

Brock: Right.

Hamilton: Women were not in that field, if you will. And I don't know if what they did you could call a field. But it was the beginning of parts of what the field became.

Brock: Right. It strikes me that both with the LGP-30 and then I suppose when you were working with the PDP-1, that you had an unusual introduction to computing from other people in that era because you had these machines to yourself. They were your--

Hamilton: Totally. Totally, yes. In fact, some things get mushed together when you look back that many years. But I also did some work in the PDP-1, again, for the same project, out at Digital. It was their PDP-1. Out at Digital in this warehouse and it was sitting on this dirt floor**,** the computer kind of like looked like it was going to fall over like the Leaning Tower of Pisa. And I was afraid to go in there because there were all these pigeons and we were all hearing about these bad diseases from pigeons, right. And so I would, I think the computer at Project MAC was busy for a couple days, so I went and I drove out there and used that PDP-1 out there. And there it was, the beginning of Digital out in Maynard. And running through the door really fast past where the pigeons hung out.

<laughter>

Hamilton: Yeah.

Brock: Also in listening to you, your stories, a lot of it's happening at nighttime.

Hamilton: Yes.

Brock: Was that your natural-- were you naturally nocturnal or was there something about the work or caring for your daughter?

Hamilton: I was very dedicated to my work, conscientious. I worked more hours than-- I mean, I was like the hackers that way. I worked a lot. But at night, I would always find a way to bring

my daughter, because it was after hours. But at Professor Lorenz's lab, I worked normal hours because I don't know, it was-- it was that way. But if I wanted to work at night, I kind of-- Well, and it was the other computer, right.

Brock: Right.

Hamilton: So that's where I would go to work. But yeah, I tended to have longer hours because I wanted to get the stuff done and I wanted to work. I was always interested in making my systems reliable as part of it.

Brock: Let's see. Well, was it in-- was it in 1961 that you started working over at Lincoln Laboratory?

Hamilton: That sounds about right.

Brock: Yeah.

Hamilton: Yeah.

Brock: Could you talk about that move?

Hamilton: Yes. Okay. Okay. I wasn't, none of us made that much money at MIT as research whatever they called us. It meant we did technical work. And it wasn't Professor Lorenz's fault, that was just the what you got when you were in that-- if you worked in the university. So I decided I'd better find a way, a job, where I could make more, because not only did I have to support the family, but I wanted to go to Brandeis, so I was saving money for that. So I saw they were advertising for programmers. I don't know what they called people, but I think they were programmers then. And this was a new thing. They were starting to hire for programmers. So I contacted them out in the Bedford, Lincoln, whatever area. I can't remember where it was. And this person, he said to me, "We're really interested in having an interview," he said, "But please forgive me." He said, "But I've only interviewed men and I do interviews in my hotel room and I can't do that with you because you're a girl." He said, "Would it be okay if we had an interview at the bar in the hotel?" <laughs> I said, "Sure, that's fine." So we had a drink while I had this interview and I was hired.

<laughter>

Hamilton: And so, anyway, so I was hired to work on the AN/FSQ-7, the XD-1 was the first AN/FSQ-7 and it was at Lincoln Labs. Okay. And one of the first things that happened when I was there, and remember, what counted was the language, if you knew the language and you'd worked on that computer. And I didn't know it, so you went to school to learn the instructions and what they did-- That's what they taught, instructions. Nothing about patterns or engineering. You just learned the instructions and then you're off on your way. So they'd started to have courses on learning the instructions. And I think it's the computer, usually the company that has the computer, that would teach it. So anyway, there was this-- Oh. And when he interviewed me, it was important for them to get programmers, that there was a shortage. So I told him I couldn't travel. I had a daughter, a little girl, and I could not go away. And he said, "Well, but we have this course which goes on. Could you at least go to the course?" It was out of town somewhere. I said, "No, but I'll make a deal. I'll learn as much if not more than all the people who took the course." And I said, "You'll see." So everybody went to the course. I don't remember, it's six to eight weeks, and I stayed back by myself at where, you know, our offices were. And by the time

they got back, I had done the majority of the work that needed to be done. And that was-- that was one of the things. And then, the first assignment that they would give to people, because they had a problem, their programmers, because people were hiring now in different places, so you had to keep your people that were already trained, right. But one of the biggest problems is everybody prided themselves on tricky programming. So if you couldn't understand what you wrote, what they wrote, they really liked that. They were geniuses, right? So this one guy, his first name was Ernie, he had written this-- He was good. He was a good programmer. And he left. And now the problem with the tricky programming is that nobody knew what it did because it was not only no comments, no nothing, but it was tricky. So when you'd come to work, it was a challenge that they'd give to the people they hired to learn this program and figure it out. So of course, that was one of my first assignments, and they gave it to me, and I just had to figure this program out, okay. So and I thought, and later on, whenever I write a program, I'm going to put comments down because this is not right what they're doing. <laughs> So anyway, I learned the program enough that over at the SAGE system, on the XD-1, I learned enough about it to try to run it, okay. And started to see, maybe I had to make a couple changes. I don't remember. It's foggy in my mind. But anyway, I got it to start running and it actually printed out, and it's kind of like, you know, when you're trying to print something that's heavy with graphics? Well, all of a sudden it started making noises like it was printing. Nobody had gotten it to print before. So we all went running over to see it, what it had printed. And it was all the comments were in Latin and Greek. In other words, this is the tricky programmer, right-- but it ran, okay. But the SAGE computer and everything around it was fascinating. Like, for example, if you had a program that-- my program was a radar program looking for enemy-- looking for noise, enemy airplanes. And if you, if the computer crashed, this is this huge computer, if it crashed, the only information you were given was what the location, the register that it hung up in with some numbers. Okay. That's all you had. So I thought-- And so what I realized-- Oh, and then, when it hung up, they knew whose program it was because there you were running your program. And all of a sudden, if it would crash, bells and whistles, really loud, everybody could hear it. Flashing lights. And then, it's like, and you're standing there like they caught you, right. The computer operators would come rushing out, the other programmers and they'd say, "Margaret, what did you do?" <laughs> So that-- So, so anyway, one of the things about my program, the one I spent the most time on, not Ernie's but my own, was whenever it ran, it sounded like the most beautiful seashore. And people would come listen to it. It was music. I didn't write it to sound that way, but it just happened to sound like a seashore to all of us. We called it the seashore program. And one night-- everything happens at 4:00 in the morning in my life--

<laughter>

Hamilton: But one night at 4:00 in the morning, I got a phone call at home and it was the computer operator. And he said, "Margaret, something terrible has happened to your program." I said, "What happened?" And he said, "It doesn't sound like a seashore anymore." So I went-- I drove in and I figured out the problem and we put it back up and everybody said, "Oh, thank God it still works," right. But it was like that kind of camaraderie between us and the computer operators. And we'd send them out to play ping pong because we liked the power of running this-- this computer and we just thought, "Oh, it would be fun to do what they're doing," and-- And then you'd go into this room and see the tracking the planes. And it was dark, but you would see the lights and colors. Just like "Star Wars." You know, it was--

Brock: Yeah?

Hamilton: It was an amazing thing to be there.

Brock: What was making-- producing the sound of the waves on the shore?

Hamilton: The computer.

Brock: Just chance or the whole--?

Hamilton: Don't ask, because I'm not a hardware guru.

Brock: But it wasn't, like, a speaker or something?

Hamilton: No, no. No. I don't know. You're asking me a question that-- It's funny, this guy who worked for me in this company, he's a professor now, said, "God, that's great. We're starting to debug things by sound now." And it had something to do with whatever the mechanics of the computer that they had-- were building in the class. He teaches computer science. But yeah, there were different sounds depending on what you ran. But it was just like when my program sounded the way it did, it just happened. I didn't make it happen.

Brock: Right.

Hamilton: Yeah.

Brock: And was your program to take the radar information and find like the signal of an aircraft in that or?

Hamilton: Well, either it's an aircraft or it's noise.

Brock: Right.

Hamilton: So we had to separate it out and find the tracking of the actual plane. And I don't remember that much about it. But yeah, most everything had to do with tracking and looking for enemy airplanes.

Brock: And was that this seashore program, did that go into the SAGE system as it was deployed?

Hamilton: In my opinion, it did. <laughs> I didn't go to upper management and say--

Brock: Right.

Hamilton: "Did you use my program?" But whatever it was, it was part of the system that we submitted. And the, here yet, was another programming language, right. And we used to use it in talking to each other. Like, "I'm going to branch left minus around the hallway." <laughs> There was a branch right. I remember the terms, because we talked the dialect, right.

Brock: Interesting.

Hamilton: We had a lot of fun. It was because we were just totally in all these jobs, including this one, total freedom. But it's funny, we were all very serious at the same time. But we had fun, you know. It was-- And there was mostly men, but there were women there too, when it got-- when everybody was hired. And more recently, I talked to one of them who said, "Oh, yeah. You were the expert." Well, she probably came six months after I was there. <laughs> But, but yeah, and we would all go to lunch at the same time. And we were like college kids, right.

Brock: Mm-hmm.

Hamilton: And we all got along. And I remember, one of the guys there, he wasn't used to having a woman there, number one, and number two, with a child. Now, I wasn't there with a child, but my daughter was home. So he said, "How can you do this? How can you work here when you have a baby at home?" And I remember, I'd not been asked that before. I said, "Well, you know, you have to do what's right for you and I have to do what's right for me." He said, "Oh, okay." <laughs> So but that's, you know, it was unusual, the whole thing was unusual--

Brock: Right.

Hamilton: For the-- for the "Mad Men" era.

<laughter>

Brock: Did the importance of SAGE to kind of like the whole Cold War <laughs> you know, defense of the nation, did that-- Did the place of SAGE within that whole Cold War context, did that weigh on you? Or did that-- How did that feel?

Hamilton: I just don't remember different feelings at that time. But I do know, you know, when I was growing up in the Quaker side of things, conscientious objectors were parts of the family. However, no doubt some members of that side of the family were in World War II and my father was in the Navy during the War. And both of his sisters, one was a WAC and one was a WAVE. So and speaking of which to do with the war, and I know I'm going off here, but I do remember another favorite book was "The Upside-Down Town." And so if you went to the other side of the world, everything was upside down. That's what the book was. So China was the other side of the world, we were told. And I remember one day in Bloomingdale, I think, Indiana, I remember digging for China. All of us were digging trying to find the upside-down town. And all of a sudden, all the bells and just like the SAGE computer, right. But the whole town bells were ringing, noises, everybody was tooting their horns. And this was 1945, was it? The War was over right then and there and we were digging for China.

<laughter>

Hamilton: Speaking of war.

Brock: Right.

Hamilton: But back in the SAGE system, which is what you're asking about, I mean, there obviously was a Cold War kind of thing going on. But I don't remember. I mean, there were certainly conversations that went on with people we hung out with. And part of what you did was you had a good time to forget the bad parts of things. I mean, you know, I mean, there was a lot of people who were self-made comedians in the group. People, I mean, they were really very bright people. But we didn't really talk about the politics that I recall. It was at home that we would talk about things.

Brock: I would imagine that that experience of standing in front of <laughs> the SAGE computer running your program and fearful of the lights and clangs and--

Hamilton: Yeah.

Brock: Bells and everything of a crash or an unsuccessful program, that would be a very visceral training in software reliability. Yeah.

Hamilton: Oh, totally. And the thing is, speaking of that, it occurred to me because one of the problems was you'd have the-- you'd have it hang up and people would write down or whatever. But we couldn't remember who was by which error. So I think I came up with the idea <laughs> of saying-- because they had a Polaroid camera there. It just had come out, right-- of taking a picture of the programmer next to his error.

Brock: <laughs>

Hamilton: Okay. So here's the SAGE computer with the register, a long register and you stand and pose with the error. Well, pretty soon, we were all getting carried away. Told the operators, "Go play in the ping pong again." And the guys would pose with a mop on his head. So we had different errors with different pictures of us. We had fun with the errors. But another thing is when we then tried to find the errors in something, I started to actually document not just my stuff but some of the stuff I had worked on. They thought that was so funny that I put comments beside the code. "Why do you put comments beside the code when you can just read the code?" I said, "Remember Ernie." The guy who-- <laughs>

Brock: Right.

Hamilton: Right. So anyway, people started to do documentation much more and maybe that became-- And then, yeah, there were certain steps we took. Not, I mean, certainly the sound helped us know where we were because it might have changed in certain areas. But yeah, reuse. We began to reuse things. Not like we do now, but it was the beginning of something other than just learning the instruction and knowing what it did.

Brock: And it's interesting. I had never really thought about that careful comment and documentation as a method for reliability for--

Hamilton: Yeah, yeah. <laughs>

Brock: But it's so--

Hamilton: And that's why I started to do it. Because I thought, "I could forget my own code, because I've been tricky. Now I'm going to trick myself," <laughs> right.

Brock: Fascinating. Well, could you talk about your transition, you know, from Lincoln into the Apollo Program and the, I guess at the time, that would have been the Instrumentation Laboratory?

Hamilton: Right. Yes.

Brock: At MIT.

Hamilton: Yes.

Brock: Yeah.

Hamilton: I found out about the Apollo Program, I'm not quite sure how I found out. I was working at Lincoln Labs in that area, so it could have been. But I think, again, it came out in the news that they had won a contract and they were looking for people to work on it at Draper. And so I thought, you know, "I guess I should delay graduate school again because I'd like to work on this program that puts all these men on the Moon." <laughs> You have a man on the Moon, right. They think the first time they'll put-- this is what it was about. So I went in and I had two

interviews. As soon as I heard about it, I went in for the interview. And two interviews from different sides of the lab. And so they asked me a lot of questions like you do, but they weren't what I was expecting. Well, one question was, what did I know about some kind of mathematics that they were into and aeronautical kinds of stuff. I said, "Absolutely nothing," right. <laughs> And so then they asked me what jobs I had in high school years and college. And I told them about some of the jobs including the Arcadian copper mine. They were fascinated with the fact I'd had this experience. And so anyway, that very same day, each one decided on the spot to hire me. Well, now I had two people offering me a job. And I thought, "Oh, my God. I know which one I want but I don't want to hurt anybody's feelings. I'll just tell them to decide." And they flipped a coin.

Brock: <laughs>

Hamilton: And the one that won was the right one.

Brock: The one that you had wanted?

Hamilton: Yeah. That went into the flight software. The onboard flight software.

Brock: If the coin had gone the other way?

Hamilton: Don't even ask. <laughs>

Brock: What was the other job?

Hamilton: Well, the other job had to do with support systems. It was software, but it wasn't the onboard flight software.

Brock: This would have been all this--

Hamilton: It was like building up for, to, you know, a facility where you did simulation on the ground. But it was only later on I thought, "I really did do the right thing. Because there's where all the drama was, right. I mean, well, first of all, I wanted to be part of the going to the Moon and landing on the Moon eventually. But boy, if I knew now-- then what I knew now, I would have just accepted that. <laughs> But I was so afraid of hurting somebody's feelings, you know. That was uppermost.

Brock: Was that, I mean, I can appreciate, you know, the thrill of that challenge of getting a person to the Moon for the first time. Had you-- was there anything in your background that may have made that vision and that challenge particularly interesting? Like had you been into astronomy or science fiction or anything like that?

Hamilton: Yeah. Well, I was into science fiction. But one of my dear friends from Earlham said, "You were always talking about going to the Moon." "I was?" I said, "In college?" And he kept insisting I was. I don't remember it, but he remembered it.

Brock: Fascinating.

Hamilton: Now maybe he just sort of went back and connected to them somehow. But, but yeah, so I was interested in physics at the time. I was the only woman in the physics class also. And at the time, I think the professor thought women should not be taking physics because he-- Well, you have to know the times, right.

Brock: Mmm.

Hamilton: So I think he wondered why I was taking the class. But I just said, "Because I want to take it," you know. That was the only time somebody in college questioned that that might not be something I would be able to make use of. But I ignored and went on.

<laughter>

Brock: Very good. Well, so can you describe for us the coin flips the right way.

Hamilton: Yes.

Brock: You know, it's software for the onboard computers for Apollo.

Hamilton: Yes.

Brock: You know, what was it like getting started and--

Hamilton: But there was all these engineers, okay, hardware engineers, aeronautical engineers and all this, I mean, a lot of them out of MIT, okay, but the whole idea of software and programming. In fact, even Dick Batten, Dr. Batten, when they told him that they were going to be responsible for the software-- and by the way, that has many different meanings, the word "software--" he went home to his wife and said he was going to be in charge of software and he thought it was some soft-- soft clothing. And the words--

Brock: <laughs>

Hamilton: It was a term that was unfamiliar to the hardware people at the time, okay. So anyway, so anyway, it was all the engineers and everything; and they were starting to work on the unmanned missions. And there's a few very interesting experiences on there. But one that comes to mind, because there are too many experiences--

Brock: <laughs>

Hamilton: Is that there was this one thing that they were worried about, what if the mission aborts. And I don't remember which mission, because I don't remember absolutes, just relatives, right? <laughs> And what if it aborted? And everybody said, "It's never going to. It just won't happen." "Oh, well, good. We'll give this one to Margaret because she's a beginner and it's never going to go there anyway, right."

Brock: Right.

Hamilton: So I wrote this program in the software that it would go to if there was an abort. And sure enough, it aborted. So it went to this program I had written, which I named "ForgetIt."

<laughter>

Hamilton: And all of a sudden, I became an overnight expert. Because they couldn't figure out what went on because I wrote the program in that part that it went to. And they called me in to find out and everything. So just like the SAGE system with its bells and whistles and sounds and lights and everything, that made a real mark on me. My God, the stuff I'm working on, this could be heard about, you know, beyond this room. And I'd better make sure-- this is on hindsight now-- I do everything in my power that would work. Well, ForgetIt worked. But the fact that it actually aborted, I mean, these guys never thought they were going to make a mistake. So we could all make mistakes, right. So anyway, then after that I-- Oh, and then another thing, I would hear these guys, all guys, right. And they'd walk around and say, "How did you solve that

problem?" And somebody else would say, "I used the augekugel method. " And I thought, "I never heard of the augekugel method. I got to find out what this is. I can't let them know I don't know the thing that they keep talking about all the time." But I couldn't find out what it was. So finally, I said, "What is this augekugel method that you all talk about using when you solve problems?" Turns out it means eyeballing in German.

<laughter>

Brock: So it's just--

Hamilton: Scanning, going through the listing, you know, understanding what's going on. So, I mean, these are just memorable, you know, that I learned if you don't know, ask. Ask questions. Don't be afraid to ask a dumb question. There is not a dumb question. What's important is to learn what you need to learn and-- So anyway, so then I started getting involved again, unmanned missions. And worrying about, you know, people started to, you know, people were working, engineers, but people began to be hired for doing programming. Well, I may have been the only one for a while. And I had all these bosses, they were all engineers, right. But then they, we started to hire. And the engineers would, I know I'm exaggerating, but they would-- we would be behind this wall and they'd throw their requirements over the wall and just expect everything was gonna start working, right, because that's what the people did back there. They made it work on the <laughs> computer, right, and so it would just magically start to work. Well, I started working on the area where you worry about how these different algorithms, once becoming software or even before, interface with each other, you know, soon I got into it because the unmanned got into the manned missions where we went from a synchronous executive environment to an asynchronous, and then who was more important and were they in conflict, and were the interfaces really interfacing? Were they communicating right? And what's going on with what? And if this guy, meaning program--I'm going back <laughs>--is using these erasables then this guy can't use them, so you've got to worry about sharing but not at the same time because it was multiprogramming, not multiprocessing in the software itself. So I got into this area which we called the systems software, which had to do with all the worrying about the interfacing and what the priorities should be in each of the programs in the onboard flight software. And so I started to learn about things like people submitting their programs, and some of the engineers that would hand their-- throw it over the wall is what we called it, they would be perfect. It would always-- they would do everything right. They'd follow our rules that we came up with as we went along the way. But others were brilliant people, but they always made a lot of mistakes. So one of the things I was responsible for in this area, making sure that they spent more time on the brilliant people making mistakes than the other people that did not make mistakes because it was much more taking advantage of the resources <laughs> of going in there. So, anyway, I concentrated on the systems software and then gradually took on in addition to the systems software the command module software. And by the way, the systems software was shared by the command module software and the LEM software.

Brock: It would be in--

Hamilton: So it would go along for the ride wherever it was going. It was like being an operating system environment at a higher level.

Brock: Got it.

Hamilton: Everybody went through it. So I remember one time I was getting a release out. We had a release every day, one for the command module, one for the LEM [Lunar Excursion Module], okay, and each had the systems software in it. So, anyway, I remember I put a change in the systems software area, might've been display routines, it might've been restart, might've been changing a priority. But I put it in in the day's release, my own coding, and everybody hung up-everybody. They were standing outside my door saying, "My program's not working. What did you do?" <laughs> The program, well, since it was a systems area everybody was hit by it. So I came up with-- <laughs> I came up with this new thing. It was called offline version, okay, so until you were happy that your change worked in the offline version, you don't put it into the main-- and that was because of the mistake I made that I decided to come up with the concept of offline versions, so it was the temp version, and then also started writing memos. It's like emails today. You might send an email to 400 people, right? You'd have to give it to the secretaries and they'd deliver it, hard copies, to 400 people.

Brock: Phew.

Hamilton: Okay, so every day it would say this went into the assembly and everything. So, anyway, I'm jumping around 'cause of my memory, <laughs> but going from being responsible for systems software to systems software plus the command module and around the Apollo 8 time, which was only using those really in flight, taking over all of the onboard flight software[12] before-- I mean just around Apollo 8 time and then taking on also-- so all the manned missions. Now I had to worry about all the onboard flight software, so it was a gradual taking on more responsibility but always keying in on everything from a systems of systems viewpoint and always being aware of everything that's going on, I mean at a higher level, not everything <laughs> that's going on. But at the level of-- I'm trying to think of the term, but being in control, of making sure that everything was in control and working together and interfacing, and so forth.

Brock: Was everyone working on the onboard flight software, were they all in the same place physically?

Hamilton: We were all in the same building, okay, on the river, okay? But now there's the onboard flight software group, which were software people, the way I used software. It's the stuff you write that runs on a computer that's the target for the <laughs> application. I have to say that because people use it so differently, especially back then. So in the group the people who were doing the systems software, taking requirements from engineers and doing application software and worrying about the integration and the like, accepting it into the version and all of that, that was a group of about--and this was my group--in its heyday around the times of Apollo 8, Apollo 11, was about 100 people, but we had 300 to 400 guests, G-U-E-S-T-S, people who submitted code, and once it became part of the assembly it fell under our responsibility. So you might have a navigation expert or a hardware expert trying to do something, but if they decided to put it into the release, they wanted to do it themselves, okay, it might be like a primitive operation we'd call it nowadays, an app or something, okay, then they had to go through our group. And each mission--like 8 is a mission; 11 is a mission--had two what we call assembly control supervisors or otherwise known as rope mothers, okay?

12 Hamilton adds June 2021: I was now in charge of all of the onboard flight software (i.e., the Command Module (CM), the Lunar Module (LEM), and the systems software onboard flight software).

Brock: Rope mothers.

Hamilton: And that was named after the fact that they eventually were shipped off to Raytheon to become frozen modules, right, hardware-ized or firmware-ized. Anyway, so all at the same time you'd have a rope mother for the Command Module for let's say [Apollo] 8 and one for the LEM for [Apollo] 8. And at the same time you'd maybe have [Apollo] 11 going on with the two rope mothers. You had many missions going on at once, and you had the systems software modules that were shared in both the LEM and the Command Module. So you can imagine the amount of care that had to be taken to make sure everything worked with everything else and that all the rules were followed. So, for example, one of the most common mistakes that was made, we realized, was that you'd transfer control from this location to something, another location, so we'd just say "TC to plus eight." Well, we had these decks of cards we'd work with and then somebody would slip a card like yourself, <laughs> and plus eight was no longer plus eight. Well, that was a major breakthrough. Change "plus eight" and give it a name, right? That's an example. So then the rope mothers, they'd look over stuff. Like the guests, the G-U-E-S-T-S, would often make mistakes because they weren't into worrying about it other than their algorithm working, in most cases, and they didn't care how it worked with the other guys. They didn't care if their priority was more or less important than somebody else's, okay? And so if their program like guidance or whatever was working, they'll say, "Oh, well, I did guidance," but you have tasks [scheduled based on time] that could interrupt them and you have jobs [scheduled based on priority] that can interrupt jobs, so it wasn't just this piece. It was this piece being interrupted by this one being interrupted by this one, and it was all a different story to the software people. It still is. <laughs>

<off-topic conversation>

<break in recording>

Brock: Okay, well, before we took our pause we were talking about the rope mothers and integrating these contributions from the various guests--

Hamilton: Yes, yes.

Brock: --and the different--

Hamilton: By the way, that's my name. It wasn't generally used. Or it might've been in the in group, "Oh, no, here comes <laughs> another guest." But, no, that was not written up as a formal term.

Brock: Right.

Hamilton: Yeah.

Brock: Right.

Hamilton: They just were not in the group. They were in another group.

Brock: With this whole process of developing each day's release, I suppose, of taking these new contributions, checking things out, the system with the rope mothers, was this kind of orchestration, was that your creation?

Hamilton: I did not come up with the concept of rope mother. I don't know-- I mean, yeah, I don't know how that name came up, but I did not invent the term. We used to use "assembly

control supervisor," but, yeah, there were other-- the rope mother might've come up definitely from one of the guys. It's the same thing, "assembly control supervisor" and "rope mother."

Brock: In terms of the overall organization of the project, was that something that was a known form for handling projects of this type or was that invented as you went?

Hamilton: Most of the things we did grew up. They were-- it's not clear what started what, but on the onboard flight software, 'cause there's all these other groups, right, like hardware groups or whatever, or simulation groups, a lot of the things we checked for we got smarter as we went along, and the way we got smarter was to make a mistake and learn from it, acknowledge <laughs> it, and say, "That's not gonna happen again," that kind of thing.

Brock: Right.

Hamilton: But eyeballing-- oh, and I should mention, too, the most famous eyeballer is John Norton, who worked on a V&V [validation and verification] company, not Northrup Grumman[13], but one of the V&V who'd take our stuff and go over it and he would "Nortonize" it. He would use the augekugel method and he would catch things as well, but we wanted to find it before he did.

<laughter>

Brock: And you mentioned for some of the-- when you were describing the kind of addressing and transfer control plus eight, that you were using punched card decks to do this work.

Hamilton: Well, actually, the programmers, those of us, we would write things by hand on coding sheets, then we would give them to keypunchers who would punch them. And I had great experience with the person who did the majority of keypunching on my stuff 'cause she would catch things. And she was a keypuncher, and she was a black woman and earning a living. I think she had a kid or two at that time, and then she went on to school and became a doctor[14]. So you could see she was gonna do something besides punching cards or being-- there were secretaries back then, too. I think we're all secretaries now, but there were people, that's what they did, <laughs> right? And now we've all been anointed <laughs> with that. But, yeah, and so the secretaries would oftentimes say, "Are you sure you want to use that word," you know, so you learned from them, right? Especially if you listened to them you learned.

Brock: Were you using one of the mainframes on campus or what was your computer for the project?

Hamilton: The AGC, the Apollo Guidance Computer, the actual computer that's gonna go onboard-- okay, several forms of it. There's the computer which could be sitting in a hardware simulation.

Brock: Right.

Hamilton: There's the computer which could be simulated by the software on a big mainframe kind of thing. And there's-- let me just see. Oh, and then we used to-- we could take the computer-- we'd run software simulations. In fact, we'd run those every night and then we'd run

13 Hamilton adds June 2021: North American.

14 Hamilton adds June 2021: Or lawyer.

hardware simulations with the actual computer sitting there, but it was the same computer that flew up there <laughs> to the Moon, the Apollo Guidance Computer, and what--

Brock: So you had one in the building. I'm sorry.

Hamilton: Yes, yes.

Brock: Oh, fascinating.

Hamilton: Yes, right. So, yeah, in fact, on the hardware simulation. We could actually-- not "THE computer," but we would run simulations where we ran one of the computers, so that's what it was. Now, you ask about other things on campus. Yeah, we'd run sims like downstairs. There's upstairs people and downstairs <laughs> people, and they're totally different people. <laughs> We never knew each other hardly, except that I remember running downstairs and really upset at them because I couldn't get some of our stuff run on the computer, and we only had turnaround once a day. So I went down there and I said, "We don't have enough time. We've got deadlines." I can still remember. And he said, "Margaret, it's only your people who are running on our computers." <laughs> It's like, so we were stepping on ourselves, yeah. <laughs>

Brock: I have read, but please correct me if I'm wrong, that in the-- at least initially in the planning for Apollo and the use of the Apollo Guidance Computer, there hadn't been a proper allotment or estimation of the amount of work involved to create the software. Is that <overlapping conversation>?

Hamilton: Oh, that's entirely possible.[15] That's the case on any software project.

<laughter>

Hamilton: So to say that, I'd say I'm surprised if there had been the opposite, <laughs> yeah.

Brock: Yeah, okay.

Hamilton: Yes.

Brock: Well, maybe we could talk about some of the greatest challenges in developing the onboard flight software.

Hamilton: Well, some people bring up the size of the computer and how many registers, and as I probably said something like that before, those are all absolutes to me. I think, do I have enough or not, right? But in other words, it was not a lot of room to do the things that we do today, but by being very careful how we design and program things, we were able to be quite clever. I guess some of it you would call tricky programming. In fact, Norton, the guy I mentioned, the eyeballing person, and I were gonna have a meeting one day, and we both were known for tricky programming at that time, but we both changed our ways over time. But you could find a way to scrunch it, right, by tricky programming. But one of the things we did which made it error prone was to have to share the erasables 'cause there's the fixed memory and the erasable memory, and you can't have two programs sharing it at the same time, and that happened a couple times, but it wasn't supposed to and it caused a lot of trouble. But, anyway, that was a way of saving but it made it more work to worry about. And then by tricky

15 Hamilton adds June 2021: In fact, this was the case.

programming did it take more time than it should have or was it something so accurate it wasn't needed, so there's all these tradeoffs that had to go on. But there was always one thing that stood out in my mind being in the onboard flight software was that it was "man rated," meaning if it didn't work a person's life was at stake if not over. That was always uppermost in my mind and probably many others as well. It had to be man rated and it had to work the first time. You couldn't fly the astronauts up there-- because it was man-machine, interfacing with the astronauts. You couldn't fly them up there to test the system, so you had to do it before the fact, meaning you had to simulate it either in the hardware or the software simulation and hope that your simulations covered everything. So you simulated all kinds of things. You could simulate the software, of course, or use the AGC. You had to simulate the astronaut, man-machine interfaces (the astronaut working with the software). You had to simulate the hardware. You had to simulate the vehicle and the world outside of it, so you had to hope that you simulated everything so it was as much like the real thing as you could get it to be. So you also had to worry that your simulations were correct from another standpoint, and actually in one case it said it didn't work in the simulation when it really did work when they tried it out in the real world, in the real world on the ground, or vice-versa. In other words, the simulation gave the wrong answers so you made the wrong conclusions. So what drives what? It's like fake news, right?

<laughter>

Hamilton: Fake results, right? Everybody says, "Oh, you made a mistake over here on the software 'cause the simulator says so." But you change the simulator-- or excuse-- you change the software and then-- in other words, you could make the crossbreeding of mistakes, so there's the single point error and you're carrying it all the way through, and I'll talk about that a little more in another-- <laughs> a little later when we worry about that kind of thing, about carrying the errors all the way through.

Brock: When did the actual astronauts come into this kind of preparation and development of the software? Were you--

Hamilton: The astronauts trained in Houston with Mission Control guys, but they also could use the simu-- they could do the simulation just-- and they did. They ran simulations. In fact, I remember when some of the astronauts came up to Draper, to MIT, and we were running the simulation, showing them how it worked, and I remember we ran the simulation and all of the sudden it came crashing down, and I realized we put the wrong numbers in at the beginning, you know. And they were watching us <laughs> run this. It's like, okay, <laughs> we can't do that again. And I think we-- I mean this is just in simulating, right? There has to be a check to make sure that we start off correctly.

Brock: Right.

Hamilton: But anything could go wrong, you know. But that's a time when we all went out to dinner with several of the astronauts who had come up to watch, and that's a whole 'nother story, but it was a very traumatic time because they got into an automobile accident coming back from <laughs> the restaurant, but that's not what we're talking about here.

Brock: Oh, boy.

<laughter>

Hamilton: I just remember it well, not just because the simulation but something more dramatic took place after the simulation, but it helped me to forget what happened in the simulation. <laughs>

Brock: Was everyone okay?

Hamilton: Everyone ended up being okay, but one of the Draper guys got beat up a little bit.

Brock: Oh, my goodness, yeah.

Hamilton: Yeah, I just couldn't-- I just thought of that.

<laughter>

Brock: Well, maybe it would be helpful to have you explain the job that the Apollo Guidance Computer did and, by extension, the software created for it just in terms of these manned Apollo missions.

Hamilton: Right. Well, basically, I would divide it probably into three parts. One part is the guidance, navigation and control, and the requirements would come in from the different experts in those areas. So you'd have "P programs" like landing. And you'd have several programs that would do these guidance, navigation, and control functions, and those were the programs, but then the astronaut would have things that he could look up--we'd call them extended verbs--and so in the middle of that you could look up values or put values in, that sort of thing. Another thing-- well, the systems software was key as to where our expertise came in, and that's putting it all together so that it would work, the applications, and then worrying about man-machine interface where you'd present displays to the astronauts and they'd come back in, so there was that interface, the man-machine interface. And then, of course, there was the whole integration part that went on where we'd have to-- with the rope mother making sure it was done correctly but actually doing it. So, for example, you'd have to decide how to-- what priority to put on all the different things that were going on at the same time, and every priority was unique. So let's say I wanted to send out an emergency and you had priorities like 10 and 20, and now you've got 20 going on because control is a higher priority than guidance, which is a higher priority than navigation, but now you want to interrupt. And also there's other things going on all at once, and you've got to make sure emergency is higher than any of them so that it can just stop everybody, because only one job could go on at the same time, but a higher priority could interrupt it and take over. So you couldn't have two software jobs going on at the same time. However, I always viewed it as they're there at the same time, so it's like they really are going on at the same time. One is just kind of asleep and the other one is awake but they're both there, so I had to worry about it, I mean me, we in the systems software, so there was all of that involved. But anything to do with the astronauts' interaction, that was the computer on both the Command Module and on the LEM, so if something went wrong, guess who <laughs> had to worry about it? Those of us in the AGC.

Brock: Well, maybe could you tell us about the first-- thinking about man rated software. <laughs> Can you tell us about what it was like for you the first time that the astronauts actually flew with your software on these computers?

Hamilton: Apollo 8 was--

Brock: Apollo 8.

Hamilton: Yeah.

Brock: What was that like for you? Where were you?

Hamilton: On Apollo 8-- well, yeah, I might get certain things a little mixed up, but Apollo 8 was very memorable.[16] It really was the Command Module that was involved on Apollo 8. But this is the one where we were running simulations, they were hardware simulations, and I used to bring my daughter in to work nights and weekends, and she would see me pretending I was the astronaut and running a mission. I would pretend I was doing what they did, so she wanted to do it, too, and that's when she actually started doing keys, watching her mother, and she started doing it and the whole thing came crashing down. The mission crashed. And I thought, oh, my goodness. I can't ignore this because this could happen in a real mission. So I checked it out to see what was happening, and she had selected the pre-launch program when it was in flight, which meant that two programs were sharing the same erasable, okay? And so I came back and told people about it, and the project managers-- we had a matrix management set up, and the project managers, one for the LEM, one for the Command Module, were the interface between us and Houston, right? And I said-- and it was like it kept worrying me and I kept saying, "We've gotta put a fix in there so that if anybody tries to select P01 during flight, if the astronaut makes a mistake, it's gonna say, no, you can't do that. You've selected P01 during flight." And the powers that be, I don't know if it was the project manager who didn't want to take it to Houston, or Houston or whatever, they didn't want to put that in because they were worried about extra code, and the astronauts would "never, ever make a mistake," quote, unquote. So I remember arguing and saying, "But we all make mistakes. Even astronauts could make a mistake." So this was this ongoing thing, and I knew what the change should be, and so I said, "Okay, can we at least put a program note in there if this should happen during flight that they can look it up and the program note will tell them what happened and will tell them what to do about it." So I remember writing the program note. I wrote it and submitted it and it said "do not select P01 during flight." End.

<laughter>

Hamilton: Well, wouldn't you know, after I was able to get that in--because it becomes part of their specs at that point or requirements, whichever--they did it. They selected P01 during flight, and that was the Apollo 8, "Oh, what happened," you know. So they then-- it was during navigation, so they had to put all the star-- I think they were putting, if I remember, "star one" they were supposed to be putting in, and they put P01, you know. So, anyway, do I remember Apollo 8? I remember Apollo 8, yes.

Brock: <laughs> So the fix was to restart the computer and then you had to send all this navigational information up?

Hamilton: Well, no, the computer didn't have to be restarted.

Brock: Oh.

Hamilton: It could even sit there and do a dummy job and nothing, and you could key in and start stuff and everything 'cause it was interactive. But they had to then start putting the data

16 Hamilton adds June 2021: NOVA documentary on Apollo 8: https://www.dailymotion.com/video/x6zndf6

back in that they had destroyed, because the pre-launch program came on and that's lift-off kinds of stuff--

Brock: Right.

Hamilton: --and it came on when you were in navigation. So, anyway, they just had to put back the data and it took a while, but right after that flight I was allowed to put that fix in there, so...

Brock: Where were you while the mission was going on, Apollo 8?

Hamilton: I was in the SCAMA room and we were pouring over the listings, <laughs> you know, these green bound books, and we were looking up-- there was the project manager and the-- there were three of us, I remember, going through it, trying to find out what happened. I said, "It's the Lauren bug. I know it's the Lauren bug. This is exactly what happened." So, anyway, it was the Lauren bug.

Brock: So you were essentially on call in your normal office during--

Hamilton: Well, this was the SCAMA room--

Brock: I don't know what that is. I'm sorry.

Hamilton: --which is the place at Draper where we talk to Houston, Mission Control.

Brock: Oh.

Hamilton: And different people come on different times. I happened-- I don't know how I do this. I happened to be at the wrong place <laughs> at the wrong time. I was there. I was there when it happened, meaning when there's drama I always happen to find myself there, but it was because I was in the area of error detection and recovery as my main-- that's the part I liked the best so I concentrated on it.

Brock: So you were able to make the fix that you had suggested previously.

Hamilton: I made the fix for the next mission--

Brock: Right.

Hamilton: --but it was the program note that immediately told us what had happened, yeah.

Brock: That's fascinating. Well, just about liking this kind of error prevention and recovery, you know, liking that--

Hamilton: Yes, right.

Brock: --I was wondering if there was-- in thinking about your interest in that, I was like, well, what sort of an interest in that, a kind of pursuit of order? And I wondered if that related to your interest in abstract mathematics or other parts of your life?

Hamilton: Well, okay, yeah, and there is a much even more interesting thing in flight, but I'll come back to what you just said.

Brock: Yes.

Hamilton: I think-- I thought about this a lot recently. I think the "what ifs" with my father and our philosophy talks were major, like, what if? What if we did this? And he would treat my

answers like I was so brilliant. I would just love to get his attention with coming up with these new ideas. I mean he made me think that I was-- like he'd never thought of it before, but I'm sure he had <laughs> many times. But he certainly-- he would listen to what you had to say and the questioning, and then he'd come up with questions, so it was almost like sitting and talking about what to put in the software to avoid a problem or-- yeah, so I think that had a lot to do with it. But, also, being in places like SAGE or even Lorenz, but being in places where if you made an error you didn't forget it, but SAGE was nothing compared to Apollo when it gets into the news <laughs> that something-- I would always think, I could always see the headlines saying, "Software crash--" whatever, you know. We were very aware of the fact now that it was man rated and it had to work, so that was-- I think to answer your question, I mean it just made you think about all the possibilities and what to do about it.

Brock: Right. You were saying there was-- you wanted to mention something else that happened in flight.

Hamilton: Apollo 11.

Brock: Yes.

Hamilton: Do you want to hear about that?

Brock: Sure, absolutely.

Hamilton: Oh, okay. Well, there's a background to this. I think it was around 1966, and I don't know what made me think of this, but I started worrying about the astronauts and what ifs, you know. And somehow it worried me, what if there's an emergency and they didn't know it? Because they're just merrily going away, reading the data and putting it in, but what if there's something really major going on and that's it? So I had a meeting with software and hardware people. By software people at the time probably I'm meaning systems people, system designers and everything, and the hardware people. And I wanted to put something in-- now, remember, we have an asynchronous environment, right, with all the software. However, we were not asynchronously communicating with the astronauts, okay? We could send something. They'd see the displays, they'd put something in, but we couldn't interrupt their displays. So what I wanted to do was to interrupt the astronauts to tell them there's an emergency[17] so they'd stop doing what they're doing, okay? So big meeting. And first the hardware guys said, "Can't be done." Remember, I'm still relatively new to this, especially to the hardware. And I said, "Well--" and, also, they all looked at me as a beginner and I'm not a hardware person, so what do I know, right? <laughs> So, anyway, I said, "I think--" so they said, "It can't be done," and I said, "Why not?" They said, "Well, first of all, the hardware is not on throughout all the mission," right. And I said, "So, leave it on. Why can't it be left on," right? And then another hardware guy said, "I don't know. We've never left it on that long. It might not work that long," right? So I said, "Well, that's too bad. Maybe we could put it on at times <laughs> when there's most likely to be an emergency." They said, "Let us think about this," right? So they came back maybe a couple of days and they said, "We've decided to leave the hardware on."

Brock: <laughs>

17 Hamilton adds June 2021: Hamilton, M., "The Language as a Software Engineer", Keynote given in Gutenberg, May 2018, at the International Conference of Software Engineering (ICSE 2018) that celebrated the 50th Anniversary of Software Engineering

Hamilton: I was so happy that-- I mean here these guys are, they're all experts. They all have their egos like everybody does, and the fact that they listened to me and they said, "Hmm, we really could--" because it was a challenge for them, right, and they came back and said, "We'll do it." Well, then the systems guys came along, "Can't do it." I said, "Why not?" And they said, "Because we've read all these things about parallel processing and what you're trying to do has a real problem because it's no longer async now. It's parallel, you and the astro-- that's a whole parallelism thing going on." So I was really upset. I got through the hardware part. Now-- that night I went home and I had to solve it because it mattered to me that I hadn't come up with something that couldn't be solved, right, as an idea to do. So I came back with a solution the next day. <laughs> And, again, these guys were gurus. I mean all these guys were gurus. They'd been around in this area for a while. And they thought about it and they said, "I think it can be done." And it was something they said had never been done in parallel processing, but I came up with this rudimentary thing if we looked at it now, because the problem was he's got his normal display and now you put up a priority display. Which one is he answering, you see? And so I came up with the idea of counting to five before he answers. So Houston-- the hardware guys got behind it. They put the stuff into the hardware, and then the Houston guys put it into their manuals, whatever you call them, checklists for the astronauts. They practiced. It was called the five-second display, so it got in all the missions starting from the landing on the Moon, so it's in there for both the LEM and the Command Module in case there's an emergency, whatever it might be. You warn them, you tell them what it is with this display, and they're given a choice. You either go here or there, that kind of thing. So, anyway, now we go to Apollo <laughs> 11 and it's time to land, okay? And so I'm standing in the SCAMA room again and they're going through all the things you go through for landing, and all of the sudden guess what comes up: 1201 and 1202 priority displays[18] [19] [20] [21] telling them there's an emergency. This is just before they land. And here were the things that I had wanted to do was to warn the astronaut when there's an emergency, and 1201 and 1202 means that there were too many things going on in the computer. One was to do with the tasks, too many tasks trying to get scheduled, and the other was too many jobs based on priority getting scheduled. So it went to a restart and the restart programs were set up to go back to checkpoints, not start the program over again, but go to the last safe place so that it could just pick up and carry on, getting rid of lower priority stuff and just-- so that's why it happened more than once. Now Houston knew, they'd seen the 1201 and 1202 before, and the astronaut knew that he had put the switch in a position that had caused extra stuff affecting the computer, and he realized, "Oh, yeah," and he put it back in the right place and they landed. So <laughs> there you have the most exciting one in my mind.

Brock: How long of a period of time was that?

Hamilton: Just seconds to spare. I mean it was just in time.

---

18 Hamilton adds June 2021: Hamilton, M., "The Language as a Software Engineer", Keynote given in Gutenberg, May 2018, at the International Conference of Software Engineering (ICSE 2018) that celebrated the 50th Anniversary of Software Engineering

19 Hamilton adds June 2021: Hamilton, M. H. (September 2018). "What the Errors Tell Us" : IEEE Software. 35 (5): 32–37. doi:10.1109/MS.2018.29011044 7

20 Hamilton adds June 2021: Alan I. Green, Robert J. Filene, "KEYBOARD AND DISPLAY PROGRAM AND OPERATION", Display Interface Routines (AKA Priority Displays): http://www.ibiblio.org/apollo/hrst/archive/1706.pdf, page 29.

21 Hamilton adds June 2021: Hamilton, Margaret H, "Computer got loaded", letter to the editor of Datamation, March 1971 (datamation_letter2.1.pdf), MIT Library. NOTE: link temporarily broken at MIT website.

Brock: And what was going through your mind during that what must've seemed an eternity? <laughs>

Hamilton: Going through my mind had nothing to do with the mission. It was like, oh, my god. My software-- <laughs> the priority displays, that was the part I had personally written, and it took me back to when the hardware guy said, "Hey, we'll leave it on." I mean I just thought of all of it. But it was the software I was thinking about and the —hardware. It was the parallel things working and stunned that it had come up. I'm not expecting-- because the switch was put in a place where it wasn't supposed to be, and I think the astronaut was used to using it to practice with because it hurried things up, but he knew right away to put it back. And Jack Garman-- I don't know if you've heard of Jack Garman-- he's the one that made the decision to go because he knew right away what the 1201 and 1202 alarms were, and he and I stayed in touch right up until this year when he died, and we talked about that often. But he'll say-- in other words, he'd tell me how the astronaut would do this. And so, anyway, but he talked about it from his perspective and I talked about it from mine, but we reminisced right up until this last June about it.

<brief interruption>

Brock: In those reminiscences, what was it like from his point of view when he saw that display?

Hamilton: Oh, he loved talking about it, and he was so humble. He would give credit to his boss and his boss' boss, and he's the one that knew the software. He was our counterpart there, meaning he trained the astronauts. He knew what 1201 meant. He knew-- oh, and his boss had said for him-- because it had happened during simulation at Mission Control just days before and even days before that, and it spooked them all because it was happening during landing. And his boss said, "I want you to write down all the alarms that could come up just in case this happens." And so Jack wrote down-- we called him Garman. Garman <laughs> wrote down-- and so they were there but he knew right away what they were, and he said, "Go, go, go,"[22] and they went.

Brock: And as you were standing there in the room in the Draper Laboratory, were they asking you for input or were you--

Hamilton: It happened too fast.

Brock: It was-- yeah.

Hamilton: Yeah. I remember looking across the room, in the SCAMA room, at the project manager, one of the guys, and I remember his face just turning white or whatever it turns, blue? <laughs> Anyway, because it was totally unexpected. But I remember this sense of relief when it landed, but it wasn't that long to be nervous.

<laughter>

Brock: And was it when they successfully landed or when they were successfully home that you and your colleagues really felt a sense of relief?

Hamilton: Landed.

22 Hamilton adds June 2021: This was just after the "1201" and "1202" priority displays presented two options to the astronauts. One was "go" and the other was "no go". Garman chose the "go" option.

Brock: <laughs> Oops, pardon me.

Hamilton: <laughs> Landed, yes. In fact, the Boston Herald on the day of the landing, front page, all over the front page, talked about that and the party we were having at Draper, so there's all these different collectibles, <laughs> if you will, at that time talking about it.

Brock: Did you just kind of go home when your shift was over or what did you do?

Hamilton: I wouldn't call it a shift. <laughs>

Brock: Or what'd you call it?

Hamilton: I would just say we were there all the time. <laughs> But, you know, we were so into always the next mission. I don't remember other than worrying about-- let's face it. We were always working on the next mission, so this was an "old" mission, but then there it is, and it was like, "No, you're not supposed to come up now. This is the wrong <laughs> time." But, yeah, it was relief, but I don't remember what we did after that.

Brock: When did you begin speaking about or using the phrase "software engineering?"

Hamilton: <laughs> Oh, yeah, this is an interesting one. Okay, you know when the guys used to throw the stuff <laughs> over the wall? It was something at Draper, Instrumentation Labs, right? And there's the engineers, hardware engineers. There were aeronautical engineers, just engineer engineers. But we were the programmers, and we were kind of like second-class citizens. I mean, you know, we'd take the stuff and we'd "make it run," right? And so in doing the simulations especially, when we were simulating everything, I kept noticing patterns between the different kinds of engineering that were very similar, because when we simulated the software or the hardware, we still had decisions to make. We had things going that were-- interrupts that had to be done. The commonality in the designs, it's like was the software this design or was it hardware? They looked alike in the simulation. But also I noticed there were things that they didn't do, like the systems software stuff. They didn't worry about who interrupted who. They worried about their stuff working and getting the right results. So when we would say, "Well, wait a minute. Which is this module? Is this hardware or software?" I said, "Why don't we call this one the hardware engineering part and this the software engineering part?" It was like I just came up with a term to say let's name this part in this whole system definition. So, anyway, the engineers thought it was funny. "There's Margaret again with her software engineering," you know. It was funny but I mean I didn't take it personally. We all laughed about it. So it was a term that had not been heard about at Draper. Had it been heard about in Russia or somewhere else? Who knows, right, but it was a term. And then there was the joking going on, saying, "Well, there goes Margaret with her software engineering," right, and one of the hardware gurus stood up in a big meeting--I remember it--saying, "You know, Margaret's right. This is--" almost makes me have tears right here. "This is engineering, what you people are doing, just as much as the stuff we're doing," meaning the hardware. He said, "She's trying to formalize it and I think it should become formalized," and everybody respected this hardware guy. And then later on in 2009, a reunion, Davy Hoag was there, and they were giving presentations for the reunion, and they'd stand up and reminisce. And I hadn't really noticed it, but Davy came over to me and he said, "Margaret, why are they just talking about hardware stuff? Where's the software stuff?" He said, "They should be getting you up there to talk about software." And I said, "Davy, why don't you tell them that?" And he said, "I tried but they wouldn't listen to me." I'll never forget that. Davy got it. I mean Davy, everybody looked up to. But he finally saw the same thing, and that

meant the world to me. Oh, yeah, this is a field, you know? So, yeah, that was in 2009. He was still alive then, yeah.

Brock: But the term, the notion, the desire is in wide circulation--

Hamilton: Yes.

Brock: --after this time.

Hamilton: Yes, yes. Well, some people come up with seeing the term being used in a program for a conference, or you've probably read some of this stuff, but NASA guys were the ones that said, "Margaret, this never was in anywhere until you started using it," and that's what I remember in my case--

Brock: Great.

Hamilton: --and we had not heard of it.

<laughter>

Brock: Let me just-- just checking our question list here. Oh, the theory of errors.

Hamilton: Yeah, I like that you named it that. That was good. <laughs>

Brock: Yeah. I liked-- I was really intrigued by just kind of like a taxonomy of errors or where they-- you know, commonalities and also a means to kind of attack them both automatically and through--

Hamilton: Yes.

Brock: --this eyeballing method**.** Could you just talk about that area?

Hamilton: My life story?

Brock: Yeah.

<laughter>

Hamilton: My life mission?

Brock: Yes.

Hamilton: Okay, so after the manned missions I guess I personally just had a sense of history about wanting to not just remember things but do something based on what lessons were learned[23] [24] [25], just like when an error would happen you'd find a way not to let it happen again, but just in general. And so first I had people in my group get all the anomalies that had been recorded throughout all the V&V, verification/validation, and find out what kind of errors were

---

23 Hamilton adds June 2021: Hamilton, M., "The Language as a Software Engineer", Keynote given in Gutenberg, May 2018, at the International Conference of Software Engineering (ICSE 2018) that celebrated the 50th Anniversary of Software Engineering

24 Hamilton adds June 2021: Hamilton, M. H. (September 2018). "What the Errors Tell Us" : IEEE Software. 35 (5): 32–37. doi:10.1109/MS.2018.29011044 7

25 Hamilton adds June 2021: Universal Systems Language: IEEE Computer Magazine Editors' Introduction Hamilton, M.; Hackler, W. R. (2008). "Universal Systems Language: Lessons Learned from Apollo". IEEE Computer, Dec. 2008.

made and how would we classify those errors. And there were some sheets that said, “Reason for error,” and it said, “Bug.” That didn’t help, right? But in most cases we could figure it out one way or the other, and we began to categorize the errors. This was funded by I think the Air Force. I got funding for this error study, we called it. And so we found out that 73 percent of the errors were what we called interface problems[26] [27]. This was during V&V, not on flight. Nothing has shown up during flight. And 44 percent were found by eyeballing like Norton or the assembly control supervisors. And I can't remember the exact percent but a very high number percent were still-- they could’ve shown up. They were still in there but we didn’t-- I don't know how we came up with that category. They were just plain scary--

Brock: <laughs>

Hamilton: --if they happened, right? So, anyway, we took those errors and came up with this set of axioms, basically ended up being a theory for systems and software ultimately, a theory of control, based upon control in the axioms, and there were six axioms having to do at the time with software because that’s what we were going back and looking at ourselves. And we came up with them trying to define a system using those axioms, because nobody understood the axioms but we did, right? <laughs> Came up with patterns, that if you use the axioms but the patterns which could be derived from them, that there would be-- the problems that we had found would not exist. There would be, for example, no interface problems if we did that. We took those patterns, which were primitives, and we found ways to build more abstract patterns in terms of the primitives so we could use the primitives plus what was derived from them, and that evolved into the language which would be based on those building blocks. We also considered that all systems inherently were distributed. They had asynchronous aspects. In other words, they all had that. Whether or not it showed up or not was something else, okay. And we also realized we could define the process of developing a system as a system itself and define that as well with this language. We began to notice really interesting patterns that came up by having the language, okay. Not only were there not interface errors, but also everything was inherently integrated. We used the theory eventually for the functional side of the system and the type side of the system. This grew into more recent times, and they were inherently integrated so that, for example, in the automation, if you changed a type now we can demote all the things using it, but it’s all formally found. You know, if the automation has access it can also find parallelisms everywhere they exist. It can find all the decisions in the definition. So now the more we studied and the more we saw these kinds of patterns, and keeping in mind things like the parent is always in control of the children, you inherently can tell a priority in this system because of the way we define it. It can automatically find out who’s more important now, okay, but we kept getting these things that we learned from those kinds of definitions. And so, for example, if you got all the interface errors out upfront and it’s 73 or more percent, first of all, there’s no need for wire tracing because there were no wire tracing errors left, okay. But at the end of the day we said, wait a minute. We don’t have to do this kind of test, we don’t have to do this kind of test, so we minimized the testing to the point where it’s the user’s intent. So if you gave it the wrong thing you wanted to do, you couldn’t do anything about it, although we could even do more about that.

---

26 Hamilton adds June 2021: Hamilton, M. H. (September 2018). "What the Errors Tell Us" : IEEE Software. 35 (5): 32–37. doi:10.1109/MS.2018.29011044 7

27 Hamilton adds June 2021: Universal Systems Language: IEEE Computer Magazine Editors' Introduction Hamilton, M.; Hackler, W. R. (2008). “Universal Systems Language: Lessons Learned from Apollo”. IEEE Computer, Dec. 2008.

But, you know, this is more later. So then it hit us that all of the languages out there in the traditional environment and the environments that go with them are concentrated on what we called an after the fact life cycle. You define it and then you do the implementation, you code stuff, you take out errors, you find better ways to test for them because you didn't catch it. After the fact are kind of like back in the old medical community where if they didn't wash their hands they had the diseases after the fact kind of thing. Or a root canal, if you'd taken care of things you wouldn't get to that point, okay. And what this was doing was a new philosophy we called "before the fact," because just by the way you define something it ended up with traceability, not just within itself in the definition, the types and the functions and the timing and everything, but also by going from the definition we could begin to automatically generate the code because it had the-- it was consistent and logically complete, okay, so now the code had that behavior, all that behavior, and now we didn't have to test for things we used to test for, okay. Did it really do what it was intended to do? That's what we concentrate on. And so we began to call that development before the fact because a lot of stuff we used to do later on was already up front. So people say, "Oh, so you're concentrating on reliability. You must do a lot of testing." No. The more reliable-- in the traditional system the more reliable, the more testing, more expensive. But, no, if you define it this way with these kinds of behaviors, the traceability, going from definition to code or within the code and everything, even instances of the mechanisms in the definition when it's executing has the same pattern because they're instances of what we call F maps and T maps. So all the way through it's got this-- it holds onto it. So we'll say, "No. In fact, the more you make use of this development before the fact mechanism, that you adhere to it or make use of it, the more reliable it's gonna be," and that went against conventional wisdom. But in fact, over and over again there were different people from DOD, from academics, government, that would have these-- Lockheed Martin called it "shootout" where we were put up against what was out there already, or "runoffs." And then in the one, the shootout one, there were 80 organizations involved in analyzing the different people that they were comparing, and they proved that-- or proved is a funny word, but they showed this stuff really did work, <laughs> and it did. It had the kinds of properties that before the fact, development before the fact had, which was traced back to this kind of behavior in the language. So, yeah, I still remember being out in Colorado. And first you had to get in and have your eyes looked at because it's a secret environment. And they had narrowed all the people in the competition down to three vendors. We were one of them. And the first-- went from requirements all the way down to the DOD life cycle, and everybody got through the requirements. But then you came to the specs and only two of us got through the specs. But now came the code, okay, and we were the only ones who could generate the code, but we had a certain time limit. So we first generated C. They want us to generate Ada because Ada was the thing. We first generated C, because it's much faster, and proved it worked. Now that we knew it worked we didn't have to worry about going through the Ada or whatever. It took longer. At the very end we were automatically generating the system, in Ada-- and the people that were observing it were saying, "Stop it. We believe you. <laughs> We believe you," and all this paper <laughs> was going up. So that was a memorable experience.

Brock: When was that--

Hamilton: Oh, let's see. When was that?

Brock: --roughly?

Hamilton: Oh, 1994? See, I'm thinking that because the report says 19-- I think it was in that time, though. And MITRE was involved, and each of the vendors had-- we had Lockheed Martin working with us where we could prove we could do development together, but they were in Colorado and we were here. Yeah, and they had these people from MITRE and from-- I don't know from which part DOD, sitting in, watching everything everybody was doing. And there's I think six volumes of this effort, yeah. So that was memorable.

Brock: Are there certain tradeoffs with this kind of development, before the fact, I'll just call it the paradigm, for lack of a better term.

Hamilton: Preventative.

Brock: Preventative, yeah, preventative rather than restorative <laughs> process, is there some sort of cost in terms of changing people's behavior or in performance that keeps it--

Hamilton: The cost is training people to do things a new way. It is a different-- it's not a traditional paradigm, so it's kind of like if somebody comes up with something new like a new language like they did with Java, it takes a big organization mostly when you're dealing with technologies, right, to make that happen. It's an investment. But I remember going up to IBM in Toronto. They had 2,000 software people working the traditional way but really good at what they did. And the person who was the head of that organization, we had a long talk about it. He said, "I'm gonna have to change the ways of 2,000 people." He said, "That's a major decision, you know?" It's that kind of tradeoff. But the biggest challenge is to do that, just that, is to educate people. But we used to train people, and we're not in the commercial mode in that way anymore, but they'd come to our facility and we would train the people, and it would be a five-day course nine to five, and so like people from Martin Marietta at the time, they were called. People would come and-- people would come who weren't even software people, and so they would all think the traditional way and want to show the same thing. Come Monday, they forgot. There was like the hump day that everybody used to call it. But, yeah, they were able to-- oh, they wrote about that in that runoff, that shootout. They wrote exactly about the learning process, everything. But, again, it's a real marketing. It's a real rolling out kind of thing, and we've stayed with the research end of things. But, yeah, I could go on forever about this part--

<laughter>

Hamilton: --because like Citibank was using it and big organizations, but they were trailblazers and they wanted to try new things. And nowadays it's a different environment. People, they use tools that are supposed to do this. They integrate them with this. It's not compatible so you've got to buy a new this one. It's a whole different world, but it was much more technically oriented. But this was real time environments. Even at Citibank it was real time on their transaction processing environment. But, yeah, so many war stories and so many great results from this.

Brock: Well, I did want to talk to you about a few other aspects of your life and career, which was, I understand that it was in the mid-1970s that you left Draper, the Instrumentation Laboratory--

Hamilton: Yes.

Brock: --and that you wanted to-- and that you indeed started a company--

Hamilton: Yes.

Brock: --at that time based on this kind of--

Hamilton: The theory--

Brock: --this error work.

Hamilton: Yeah, yeah, the error work. I like that.

Brock: <laughs> Could you talk about why you decided to leave at that time and about your first company?

Hamilton: Yes, that was a company where we were doing applications using our theory for DOD, mostly DOD. Yeah, so we were in charge, no investors. But we had I think it was the Rockefellers, Venrock, but they wanted to come to us to talk about investing, and so we brought in the-- and they were nice guys, okay? But then they brought-- you know, one thing led to another, many investors, and now we're getting to be over 100 people. And, you know, we had sales forces all over the world but not huge, you know, not like some of the companies that are huge. But for what we did, it was all concentrating on our language-- I mean our theory, but we hadn't gotten to the point where we had USL. This was earlier stages of the things that we did and we were concentrating on software, not systems, not a whole-- not development before the fact. But, anyway, the investors came in and they felt strongly that we could make a lot more money if we left the-- it was like what I called the Digital world, because we had several, many customers using our product that was based on real time kind of work, engineering. So, anyway, but they came in and they-- oh, I forgot about an important thing. Yes, okay. Ross Perot's company--

Brock: Oh.

Hamilton: Yeah.

Brock: EDS?

Hamilton: EDS, yes. They came up for a meeting--they had heard about our stuff--and they asked me how much-- and they were impressed with our work, and they asked me what our company would be worth and would I consider selling it. And I still remember saying, "I wouldn't take anything under a billion." And the thing is <laughs> back then a billion was a lot of money, right?

Brock: <laughs>

Hamilton: Well, they thought that was-- they knew that I believed <laughs> in what we were doing and everything, and we worked with them for over a year back and forth negotiating and everything, and we had a deal. They were going to invest in us-- excuse me, they were gonna buy us. And I think it's either 37 or 38 million, which at the time was huge, okay? So a deal was made at a board meeting, handshakes. All of the sudden, EDS was being wined and dined by General Motors, who was buying them out and said they had to stop all acquisition efforts. And I was dealing with Mort Meyerson, who was the CEO under Ross Perot, and somebody actually asked him if he remembered that amount, and he said, "No, it wasn't 37 million. It was 38." <laughs> But, anyway, so at this time we had many investors and they thought this was gonna take off, right, for them, I mean that this deal was gonna go through and we'd support them and

work with them and everything. But all of the sudden the deal was broken, and so they had to blame somebody, so maybe it's the marketing guy's fault, right? But ultimately it fell on me because I'm the one that negotiated that work with them, and if I had done it faster this wouldn't have happened to us. <laughs> So, anyway-- and it's probably true, but in these days a handshake would've legally meant something. Back then apparently it didn't in hindsight. I'm looking back on it. So, anyway, they then brought in new management and I became chairman of the board. I was anyway, but I was-- I guess they call that being kicked upstairs.

Brock: <laughs>

Hamilton: I know-- well, I was CEO, but then they brought in a new CEO. So everybody there decided that the important thing was to go commercial, meaning IBM. Our product was not on the mainframe, on IBM. It was on the VAX. And by going in that direction, we were basically gonna-- our customers we were leaving behind, and that was off and running, really seriously off and running. But there was-- yes, IBM had loads of machines and you could make money and everything, but it wasn't the path that we were on. We were like a sailboat and they were like a big ship, right? So, anyway, they even came up with an idea for a product that was doing some of what we did in our product but it was simplifying it, and, by the way, they could just market it and sell it, but the technical people weren't gonna be-- the people who were developing were being left behind, basically. So I decided it was time to go home, <laughs> in other words, to leave that company and think about taking our technology and evolving it further, and so that's what happened. I think we departed ways in as friendly as one can, and I decided I would just go to this part of Italy and become a waitress and forget about this entire world. But then some people came to see me that were friends of mine from DOD, and they said, "Margaret, you're not gonna go over there. You're gonna keep going with your work and we're gonna give you some funding."

<laughter>

Hamilton: And so we were off and running again and going from the technology, which was concentrating on the functional side, the doing side, the time/space, to including all of the above, okay, and so we then began to work on expanding it to include everything that one could think of in a system and came up with USL.[28] [29] [30] [31] [32] [33] And then we came up with 001, which was its automation, and that was maybe 30 years ago.

---

28 Hamilton adds June 2021: Universal Systems Language: IEEE Computer Magazine Editors' Introduction Hamilton, M.; Hackler, W. R. (2008). "Universal Systems Language: Lessons Learned from Apollo". IEEE Computer, Dec. 2008.

29 Hamilton adds June 2021: Hamilton, M., " Inside Development Before the Fact", Electronic Design, April, 1994, ES

30 Hamilton adds August 2021: Universal Systems Language (USL) and its Automation, the 001 Tool Suite, for Designing and Building Systems and Software, Lockheed Martin/IEEE Computer Society Webinar Series, Margaret H. Hamilton, Hamilton Technolgies, Inc. September 27, 2012

31 Hamilton adds June 2021: Hamilton, M., " Development Before the Fact in Action ", Electronic Design, June, 1994, ES

32 Hamilton adds June 2021: Margaret H. Hamilton and William R. Hackler, " A Formal Universal SystemsSemantics for SysML ", Seventeenth International Symposium of the International Council on Systems Engineering (INCOSE), June 27, 2007.

33 Hamilton adds June 2021: M. Hamilton and W.R. Hackler, " Universal Systems Language for PreventativeSystems Engineering ," Proc. 5th Ann. Conf. Systems Eng. Res. (CSER), Stevens Institute of Technology, Mar. 2007, paper #36.

<laughter>

Hamilton: But we're research people now. We have concentrated on that.

Brock: On pushing--

Hamilton: Not the commercial part. We've turned over certain work to people who have been customers, but we're concentrating on just getting it recorded, getting it out there, and putting things in it that carry out our most recent ways of making it even more streamlined in USL.

Brock: What happened to that first company?

Hamilton: Well, okay, I left-- I'm trying to think of when I left, but it was at least two years later, maybe three years later that they folded.

Brock: Oh.

Hamilton: And, yeah, but I didn't know-- I mean I wasn't involved.

Brock: Right.

Hamilton: It was just like two different companies. They went in a totally different direction. They had only the IBM product, and we were still selling and dealing with customers who were in the VAX environment and then the Linux environment, so we stayed true to our beginnings. <laughs>

Brock: And in your present firm, Hamilton Technologies, with that research-- as a research company, if you will, has that been through getting mostly research contracts from the government or has it also been from industry?

Hamilton: Yes, <laughs> even academics but, yeah, I mean when we were still concentrating on selling the product. It's all of the above but probably more government, but also like Citibank was a customer. Scott Paper was a customer. But they're, again, the real time world, the engineering side of the world that were mostly involved.

Brock: Right.

Hamilton: But, yes, if somebody were commercially oriented, yeah, I think they could do some very interesting things with it as a commercial endeavor.

Brock: Well, just being mindful of the time, I thought perhaps-- it's four o'clock. I thought perhaps I could switch to the set of more general reflective questions at the end of the question list--

Hamilton: Yes.

Brock: --if you feel that we've hit the-- that we've done a good enough job <laughs> of--

Hamilton: With the questions. Do you feel like--

Brock: --of covering the-- I think I would like to ask some of these general reflective questions, so maybe if it's okay with you we could just turn to them.

Hamilton: Yes, okay.

Brock: Okay. One, which is a question which would personally frighten me to try and answer <laughs>--

Hamilton: Yes.

Brock: --is just about proudest moment of your life or one of the proud moments of your life.

Hamilton: Yeah, I don't think I'm comfortable with the word proud--

Brock: Hmm.

Hamilton: --but the most memorable, the most exciting, the thing I think back to probably more often is that whole experience with Apollo 11 and finding a way to work within a distributed environment coming out of a synchronous to an asynchronous multiprogramming, and the fact it was a distributed-- in other words, it's the discoveries and the breakthroughs that are the most exciting to me. Yeah, that's-- I just can't forget them.

<laughter>

Brock: And this, I think, is another kind of interesting question. The real turning points in your life that led you to the career that you've had and maybe these inflection points where your life really may have gone in a different direction, if you have reflections on that.

Hamilton: Yeah, I think probably when I think back, probably turning points were having to struggle and make a living in my own family where, because I had to go out and work and do things and I had to kind of be thrown into situations, even moving many times, where I had to deal with unknowns all the time, unknown people, <laughs> you know, learning how to be around people from different places, being around unusual father, unusual grandfather, Florence Long, but just the influences I had along the way. Professor Lorenz, definitely. I think back to him now more than I ever did before because I realize how much he taught me without trying to but just his influence and how even just looking at understanding something because of the coffee, how he looked outside of his world and came up with new things, right? So that was a turning point, but also the fact that I had to go out and make a living, which got me into things I wouldn't have gotten into. If I weren't out there working to try to get money for graduate school or support the family I maybe would never have gotten involved in the things that I got into. But the variety of jobs, the variety of things, the variety of places, I mean variety is--

<laughter>

Hamilton: --as they say the spice of life, but certainly that was part of it. So turning points, how do you go back and make one turning point? It was a collection or an integration of things, right?

Brock: In some ways we just answered the next question on this list, was role models and people who you really admired and looked up to. It seems like you've--

Hamilton: Like I <laughs> included that in there, right? Yeah, my father, my grandfather, my grandmother, who was a journalist, and Florence Long, the mathematician, Professor Lorenz, along-- I mean it's really important. People in your life can really influence what you do, which kind of scares me 'cause I hope <laughs> I influence my family in the right way. But it is amazing.

Brock: Yes, absolutely.

Hamilton: And, actually, the younger you are probably has more influence, as they seem to think that this is the case. So, yes, going back to the family years and being around at times like the war, when the war was over and the impressions that it makes and the books that you read.

Brock: Hmm.

Hamilton: You know, "Dick and Jane," <laughs> right, all the way from "Dick and Jane" to "A Thousand Years of Solitude."

<laughter>

Brock: The next question is about any life lessons or advice that you'd like to impart to young people, and maybe young people considering a career in doing something technical.

Hamilton: Yeah, you know, I have always found when I've hired people the combination of the experts and the young kids works best because sometimes the experts can get stuck in a traditional way and the young kids might come out and say, "Why this," right? And I think I've learned along the way from the young kids. But keeping in mind there's old people that are still young kids at heart, okay? They have an open mind. But I guess don't be afraid to question things and don't be afraid to ask so-called stupid questions. I mean I remember-- this is a little off from your question point of view, but thinking in different ways, you know, like the sour milk or something like that. But at Earlham, believe it or not, in order to graduate you had to do a somersault in phys. ed. and I could not do a somersault, and I thought, I'm not gonna graduate. I haven't passed this, right? And so for years I'd not been able to. I was just afraid of this thing. And all of the sudden-- and I was taking ballet, swimming, and all that, and it hit me. I can do a somersault. I'll do it in the water. <laughs> So I passed that physical ed. exam. Well, it's learning to think of solving a problem. If you can't solve it put it in a different place, you know, and don't be afraid to disagree with the experts. You know, in our company I'd always say never say never, yeah, and never give up. Just because people say it's never gonna work, you know, that doesn't mean you have to give up. And there have been many times when people say things like that and you ignored them and it was a good thing.

<laughter>

Brock: And this is a-- I would love if you have any thoughts about what the future may hold for software or maybe for computing more broadly. This is a question that I thought of actually last night. <laughs>

Hamilton: The future for software. Well, first of all, I guess the before the fact paradigm has a big chance I think in the future of taking off not because it's an elegant way of doing things perhaps in some people's minds, not even because it costs less to develop it 'cause you don't have to do as much work along the way. Well, maybe because of that, because it would save a lot of money if you did it that way, but it might cost a lot to get people educated and turned around, but if you're dealing with a paradigm or a language in an environment which can handle any kind of system, then maybe some of the problems in AI that could be there because you're using earlier paradigms might speed up more because, well, some things are just plain too expensive to do, and if you can save a lot of resources to do it a more modern way, then you might solve problems you wouldn't have solved before 'cause it was too expensive to try to solve it, like going to Mars or whatever. So I think there may be more consideration of doing things in a more modern way and not just keep doing things 'cause that's the way everybody is doing it. Like, you

don't have to do a somersault the way people think somersault, right? <laughs> But, yeah, thinking, as they say, outside the box or some people, never in the box.

<laughter>

Hamilton: But, yes, I think what I'm talking about is the science of software, the-- what's a good term for what I'm trying to say? The mathematics of systems and software and how it affects automation and the way we build systems and whether we're even able to do it from a breakthrough point of view or from a cost point of view.

Brock: Hmm.

Hamilton: But I see it heading in that direction if I have anything <laughs> to do with it.

Brock: Well, Margaret, maybe we could talk a bit about, um, just after Apollo and the development of your work, the error studies, and also how that work related to your efforts with both Skylab and the Space Shuttle.

Hamilton: NASA continued our funding because of Skylab and the Space Shuttle, and Skylab was very much a continuation of the work on the onboard flight software for the Moon missions. On the Space Shuttle, we were asked to give requirements, so we spent a long time making recommendations for the Space Shuttle; for example, how important it was-- 'cause we'd gone from a synchronous environment to an asynchronous environment to build our onboard flight software. Asynchronous software was much safer, we thought, than the synchronous software. So that was a recommendation, as an example. But during this time, the Navy got wind of the axioms, the six axioms, which was this formal mathematics for software, okay, and so the Navy called and asked for me. And I had some people in my group that were in my office at the time. And they wanted to speak to me and they said, "How would you like to build a specification language based upon your axioms?" And I said, "I would love that. I would love that." So I got off the phone and I said to everybody, "You know what? I don't know. What do they mean by specification <laughs> language?" I said, "I'd better call them back and tell them I don't know what they mean," so I called them right back and I said, "What do you mean by specification language," and they said, "We don't know. We were hoping <laughs> you would know," so we got the funding for the specification language based on that early axiomatic theory. And then the Army called up and said, "How would you like to define the development process as a system and build an automation based on that?" So then we started working with the army. And in the meantime, NASA is involved throughout much of this, so we had many-- oh, yes. And then Draper was very intrigued by the axioms, and so the IR&D people would come up I think once every four years to decide which research was the most valuable for industry. So they came up from the different agencies, okay, Army, Navy, Air Force. Eventually people became the Star Wars people. And so Draper selected I think it was six research types from Draper, and what we'd been working on was axioms. And then there were others there and they had questions that we had to answer in our presentation. And I'll remember that one of the questions was, does this apply, does your work apply to DOD? And everybody would say how it applied and everything. And I remember creating this slide with huge colored "yes" going across the slide.

<laughter>

Hamilton: And that seemed to win them over a little bit, my enthusiasm. And then I was going through the axioms and explaining what it meant, and I remember Cliff Mclain, who was very

good friends with the President Duffy at the time, and one of the things that's in there is because of the axioms. If you produce an output somebody's got to take it in as an input. It can't just be hanging out there. And if you have an input somebody has to pick it up and do something with it. And Cliff said, "Yeah, we have a few people like that that could do something <laughs> about that." So, anyway, we all got graded like, you know, I don't remember, 4.0's, like grading school in college, right, whether it's 3.1-- whatever, and this was for 6.1 research kind of efforts. And so I got totally across the board-- 4 is the best-- all 4s, and they were all excited about it. And then that's when Cliff said he wanted me to come down to-- I think he was the head of civil defense at the time or what became eventually civil-- not civil defense, but to do with Star Wars, the pre-Star Wars effort. So he said, "I'd like you to come down and have a fireside chat with me about your axioms," and I thought, oh, okay, I'll go down and talk with him. There'll be a fireplace and that'll be very interesting. I go down there and there's this huge table with all these experts <laughs> for this fireside chat. And then we got our funding, I think it was funding for civil defense where we were to define civil defense itself as a system, not software system, but civil defense itself as a system, so we did that with the language in its earlier stage. And we got a call from the CIA saying, "How did you know this," right? By defining civil defense as a system it showed inputs, outputs, and it led them over to other places where they got data or information and they were places we shouldn't even have known about, but by actually interviewing people and questioning them we would learn where it came from. So we didn't know about like the CIA, but we discovered it was coming from somewhere and then people could-- in other words, so by actually formally defining a system you learn more about it than the people who even built it, and that kind of thing happened even in a manufacturing place down in Alabama where we found out-- I mean they found out that when they talked about making changes they didn't consider things like decisions in all of their decisions. <laughs> So they found out much more about where the work was being done, where they were wasting-- and it was defining the shop floor all the way up to the management as a system. It could've run as a piece of software, if you know what I'm saying.

Brock: Right.

Hamilton: So we began to use it then for systems, not just software.

Brock: In listening to you talk about that, I am reminded a bit of cybernetics and that-- not to say the exact same approach, but the interest of looking-- it's kind of a science of systems, you know, of interactions, you know. I guess decisions is a form of feedback, you know--

Hamilton: Which all systems do inherently, even those not explicitly shown, right--

Brock: Right.

Hamilton: --if you think about it.

Brock: Was that a connection that you and your colleagues drew to that earlier whole literature about cybernetics?

Hamilton: Meaning when did it begin?

Brock: Like Norbert-- oh, that would've preceded your work--

Hamilton: Right, right.

Brock: --you know, in the forties and fifties, I guess.

Hamilton: Right. No, we didn't really-- there was-- I'm trying to think of the name[34] of something that it reminded us, our customers, researchers of. I'm trying to-- it's a mathematics, but it didn't have all of the real time in it. In other words, the mathematics that we had come up with has the control aspect in it[35], which considers access rights and things like that, yeah, and the prioritization of things which considers importance[36], so it was a different dimension onto any mathematics that was around--the control part--so access rights, for example. Most important is controlling--who's allowed to invoke who--you could only have one parent. None of this had we or people found, but there's a mathematics I'm trying to think a name of which did some of it, the functional side part of it, but it didn't tie together the entire-- things like parallelism.

Brock: Right.

Hamilton: I'm trying to think of some other aspects that we learned along the way, and there were no rules as to where you could go get stuff from[37]. I'm thinking of the axioms now[38] [39]--

Brock: Yes. <laughs>

Hamilton: --as I'm going through it. <laughs> It's like, or detecting an error when it's running in real time (the axioms). You look at something and it won't let you go any further, anything that the automation could now do and "take off with".[40] In other words, if you could automatically go from the definition and actually run it in the real world, anything that would make that be able to happen inherently or automatically is where the aspects of control helped solve that problem.

Brock: I guess my last, last question for you, <laughs> if that's okay--

Hamilton: Yes.

---

34 Hamilton adds July 2021: Category Theory. S. Cushing, "A Note on Arrows and Control Structures: Category Theory and HOS, Candidate BMD Data and Axioms," contract no. DASG60-77-C-0155, HOS, prepared for Ballistic Missile Defense, Advanced Technology Center, June 1978.

35 Hamilton adds July 2021: Based on the 6 axioms of control of our theory. Hamilton, M. H. (September 2018). "What the Errors Tell Us" : IEEE Software. 35 (5): 32–37. doi:10.1109/MS.2018.29011044 7 M. Hamilton and W.R. Hackler, "Universal Systems Language for Preventative Systems Engineering," Proc. 5th Ann. Conf. Systems Eng. Res. (CSER), Stevens Institute of Technology, Mar. 2007, paper #36. This paper discusses USL, the universal systems language that takes some of its early beginnings from the original theory, in particular as it pertains to aspects of control.

36 Hamilton adds July 2021: M. Hamilton and W.R. Hackler, "Universal Systems Language for Preventative Systems Engineering," Proc. 5th Ann. Conf. Systems Eng. Res. (CSER), Stevens Institute of Technology, Mar. 2007, paper #36. This paper discusses USL, the universal systems language that takes some of its early beginnings from the original theory, in particular as it pertains to aspects of control.

37 Hamilton adds July 2021: Category Theory. S. Cushing, "A Note on Arrows and Control Structures: Category Theory and HOS, Candidate BMD Data and Axioms," contract no. DASG60-77-C-0155, HOS, prepared for Ballistic Missile Defense, Advanced Technology Center, June 1978.

38 Hamilton adds July 2021: Based on the 6 axioms of control of our theory. Hamilton, M. H. (September 2018). "What the Errors Tell Us" : IEEE Software. 35 (5): 32–37. doi:10.1109/MS.2018.29011044 7

39 Hamilton adds June 2021: M. Hamilton and W.R. Hackler, " Universal Systems Language for PreventativeSystems Engineering ," Proc. 5th Ann. Conf. Systems Eng. Res. (CSER), Stevens Institute of Technology, Mar. 2007, paper #36.

40 Hamilton adds July 2021: Anything that the automation could now do, and "take off with" (running); if it comes across invalid data, it won't go any further (and identifies the Reject value), because of the six axioms.

Brock: --would be, driving here today I was listening to a podcast, and the subject was about essentially diversity issues of all kind in the tech industry today, and this was focused on the Silicon Valley context. But the discussion was about both people of color and women, and--

Hamilton: How did I know? <laughs>

Brock: Well, sorry, it just occurred to me.

Hamilton: <laughs> Yeah.

Brock: And I hesitate to ask, and we can skip it if you'd like--

Hamilton: No, no.

Brock: --but, you know, I just wondered if you had any reflections on that sort of-- the continuing what appears to be a struggle for gender equity, gender diversity, particularly in software but in technical fields more generally.

Hamilton: Right. The reason I said how'd I know, because I am asked that--

Brock: Yeah.

Hamilton: --more often in these days than in earlier days.

Brock: Hmm.

Hamilton: But maybe it was more in the earlier days everything was right out in front as to how people felt about-- you know, the whole Mad Men kind of thing, for example, era. But I've been a-- I gave a talk at my college, at Earlham, what they wanted to know, what is it like being in the engineering field and being a woman, so I talked about that for one of their special colloquiums or whatever it was. And I was amazed that afterwards people would come and they'd want to talk about it. But there were professors all over the place who basically said it was a problem, and I was surprised that it almost seemed to be worse in some ways today than it was back in the early days because then you were sort of like just an exception because you just sort of were doing a good job so they decided to give you more, right?

<laughter>

Hamilton: But I mean there didn't seem to be that much of a distinction. You know, but now I do remember on hind-- it's made me think more about what it was like back then, and women were paid less. And I remember several things back then. I found out that some men were making twice as much as women for no reason, but excuses were given out like, well, because women get married their husband will support them, so they don't need as-- I mean these were actual reasons that were given. And then I remember having somebody that worked for me trying to take a loan out from the credit union, and they said she needed her husband's signature. This was when I was at MIT. And I said, "Well, would a guy need his wife's signature? No." So I went to the credit union at MIT and I told them, "This is not fair." And the credit union was made up of mostly men but some women, and the men all agreed with me that the rules should be changed. And a couple of women said, "I know I think my husband should approve this." And I said, "Well, you can go ahead and have your husband approve it, but I'm not gonna make it for every other woman. That's not right." So, anyway, they changed the rules. And so I don't remember which question you asked me in this way, but I do know that I think it's tougher now in some ways because of things like the internet where bullying is easier and it's hidden. I mean

if things were happening back in the old days you could kid back. You knew who they were. Now you can go into things like-- you know, they'll write about somebody on Wikipedia and you'll go back to novelists, and they went from being one of the best novelists to being one of the best women novelists. And you don't know who's changing it because-- but I'm saying I think there's a whole new set of problems, and bullying can happen when you don't know who's doing the bullying online. I'm not an online person myself. I don't join any of this stuff, but it's a new world, and so how do you fight it? We have to understand it. It's like any enemy, right?

Brock: <laughs>

Hamilton: You have to understand your enemy. And I think people are getting by with things that are probably making maybe-- this is just my humble opinion, that may be making things more difficult for minorities and women, and so you have to hit it from-- like when I had to fight the-- I did several things like that on hindsight because it wasn't fair. I didn't think of it as men and women. It wasn't fair. But I think about it in our culture and I think it's a really cultural problem. When you see women not being allowed to drive in certain countries or you see women can't become priests, and you start seeing it-- and you know how I feel about system of systems of systems and the butterfly effect or whatever. Every single one of those things impacts our culture or impacts women or minorities as to whether they can even do something or not, 'cause kids might think, well, I can't do that because I'm this or I'm that. Until we start making changes, until our leaders stop admiring people who do things that encourage that, we have a problem, right?

Brock: Yeah, absolutely. <laughs>

Hamilton: And so, yes, I think it's a cultural thing, and the only thing we can do as individuals is take one thing at a time and take somebody under your wing and try to help them do the work. So in my case, for example, I said, "Can you help me?" In one case, if somebody's working for you, you should be able to make at least as much as people <laughs> that work for you, that kind of thing. And then you get somebody who is open minded-- not open minded--who gets it-and you get them to work for you sometimes because you know they can do it better because they're not the minority. So you work at it from bottom up and from the top.

Brock: Well, great. Well, thank you. Well, I think that's all the questions.

Hamilton: Yes.

Brock: Thank you for that answer.

Hamilton: Okay, great.

END OF THE INTERVIEW

