Stanford eCorner

Solving the Software Crisis

23-10-2019

URL: https://stvp.stanford.edu/clips/solving-the-software-crisis

In the late 1960s, after writing a thesis that applied AI to chess and earning her PhD in computer science at Stanford, Barbara Liskov returned to Mitre Corporation. Within a few years, she found herself doing research related to the "software crisis." She describes how, while working on her Venus operating system, she was able to divide a computer program into smaller, discrete units. That very practical solution became a fundamental concept that guides how computer programs are built. She explains how accepting a faculty position at MIT then allowed her to fully devote herself to the problem of programming methodology.

Transcript

- So I finished my PhD in AI.. It was on a program to play chess endgames.. And in those days chess was a, sort of, a killer app for AI.. The reason was that computers were not very powerful, and they just could not do the kind of search that's needed to figure out which move to make.. And so, that was why I was working in that area for my PhD topic.. Anyway, I finished my PhD and nobody was willing to offer me a good job as a faculty member.. And so, I ended up going back to Boston and working for the same company that I had worked for originally, MITRE Corporation.. It was all my husbands fault because I wanted to move back to Boston because he lived there.. So I got to MITRE, and it was actually very fortunate that I did not go on to a faculty position right away because it's not easy to start up as a new member of the faculty, and at the same time, you're teaching courses.. And you have all these obligations, and you're also trying to change fields..

And, I was making a major change of field from AI into computer systems.. And, being at MITRE gave me the freedom to do this.. I was in a research position now, and the first project I worked on was a time-sharing system, which time-sharing was a hot topic at the time.. And, I worked on that for a couple years.. And then, I was at MITRE, and they do research for the government, and I was asked to look into this problem that the government was interested in.. Namely, what to do about the software crisis.. So, the software crisis was, people would build big programs, and they wouldn't work.. They'd spend hundreds of millions of dollars, hundreds of man years, and in the end they'd have to scrap the whole thing.. And actually, in the 60s, the 70s, the 80s, you could read in the newspaper about these fiascos.. Company such-and-such, you know, spent all this money, and now they've had to throw the whole thing away..

So, the software crisis was a really big problem, and I was asked to start thinking about this.. And so, I started to look into this field, it's called Programming Methodology Of course I read all the papers that existed, and there were some really good people working in that field.. There was Ed Stradykstra, Tony Hoare, Dave Parnas I mean these were very good people, and they were writing papers about, "How do you break up a program into pieces, so that you can reason about it?" The problem that they were worried about was, software programs are huge.. They were huge then, you know? Millions of lines of code.. They're even bigger today.. There's no way that you can make sense of something that big.. You have to have a way of breaking it up into small pieces that you can work on independently, reason about independently.. And then, somehow, you put the whole thing together and it works.. And nobody knew how to do that.. So they were talking about things they called modules, but Parnas said, "I don't know what they are..

There's these things called modules, but I don't know how to describe them." And, they were also worrying about how do you do design.. And, Nicholas Veriko wrote a paper about top down design, and he sort of talked about it but it was unrelated to the software structure that existed underneath.. Anyway, I read all these papers and I realized that I had invented a software methodology when I was working on my first project, The Venus System.. Because, that was a complicated project at the time and I was very worried about how my small team of programmers would manage to build all that software and have it work it a short period of time.. And, so what I did was I sort of broke the rules in the way that programs were being built at the time.. In those days, there tended to be lots of global variables and then, lots of code.. And, the code interacted through the global variables.. And, that didn't actually work all that well so, what I decided to do was to say that I was gonna not have any global variables.. I was gonna partition them into discrete units and there would be some code responsible for each partition.. And, the only way that code can interact with one another, in that, you can get access to the globals, was by calling operations that the partition that was in charge of those globals provided for you..

So, I had this notion of what I called a 'Partitioner', a multi operation module, that had some data hidden inside and a bunch of operations that you would use to interact with that.. And I wrote this up.. - Which is, sort of, a fundamental concept you encounter in a CS 1068 class.. - Well, we hadn't quite got there yet.. - (laughs) - So, this was at MITRE.. So, here I am at MITRE.. And, meanwhile, I had written a paper on my operating system, Venus.. And, I submitted it to SOSP, which is the top conference in computer systems.. And, I presented it at this conference and, unknown to me, there were people from MIT kinda looking for women.. So, what had happened is, Title Nine was on the verge of being passed..

And, Title Nine, although it has to do with "Athletics", actually started to open the door for women.. And, the President of MIT, Jerry Wiesner, I think was interested in having them hire some women.. And the Electrical Engineering Department, which was all that existed at that point, had gotten the message.. And, so, they were kinda looking.. And the chair at my session was a professor at MIT, and there were a couple other senior professors in the audience.. And, as a result of this talk, they asked me to apply for a faculty position.. And, so, I moved to MIT in the fall of 1972.. And, this was actually a really good time to make this move for me because, at that point, I was totally wrapped up in the programming methodology question.. In particular, I wanted to understand.... What can we do to help people figure out how to break their problem, when their doing design, into a bunch of modules that make sense? And, nobody knew how to do that..

And, the benefit of being a professor was that I get to find my own research direction.. And I had a research direction that I was really interested in pursuing...