

Medford OR  
1:29:31  
m4v

Nils Nilsson

An interview conducted by  
Peter Asaro

Jun 14 2011

**Peter Asaro:** Your name, where you were born and where you grew up and went to school.

**Nils Nilsson:** My name is Nils Nilsson. I was born in Michigan in a town called Saginaw, Michigan. Lived in Michigan until I was eleven years old when my parents moved to California. I went to high school then in southern California in Glendale and then onto Stanford. So I did all of my work at Stanford, both my undergraduate work and my PhD work.

**Peter Asaro:** And what did you study as an undergraduate?

**Nils Nilsson:** As an undergraduate, I majored in electrical engineering and with a specialty on information theory, so information theory was rather new then. I think Shannon's book came out in the early fifties. I started thinking about-- or maybe late forties. I started thinking about information theory as a first year graduate student in 1956 and finally did a dissertation not so much on information theory, but one that involves statistical detection theory, which is related and that work involved looking at the theory of games and how it could be used to design optimum strategies, both for a radar operator and someone who wanted to jam the radar. The optimum strategies involving the way to spread the bandwidth around, how to spread bandwidth, so that was my thesis.

**Peter Asaro:** Who was your advisor?

**Nils Nilsson:** Willis Harmon. He's passed away, but Willis Harmon, as a matter of fact, did a lot of interesting work there at Stanford, some of it not related at all to technology. He was one of the very early people who was intrigued with what the effects of LSD might be and he thought it could be used as a sort of curative agent for alcoholics and people with various psychological problems. That was before LSD was outlawed.

**Peter Asaro:** Was he in the psychology department?

**Nils Nilsson:** Willis Harmon was an electrical engineer and he had learned about information theory himself in Florida, I think, where he got his degree and then he came to Stanford and taught courses in it and I thought, "Gee, this is all pretty exciting stuff." I was never very good at experimental electrical engineering. Although I had several experimental laboratories, I always had problems getting the laboratory results to work out the way they were supposed to. There were always these hidden ground loops and things like that, so when I ran across theoretical things, I thought, "Well, that's a lot cleaner" and it appealed more to me.

**Peter Asaro:** So that was your Master's thesis?

**Nils Nilsson:** It was a funny situation at Stanford. I never got a Bachelor's degree. They had a special program at Stanford to go straight for a Master's in five years and the reason for that was that without having a Bachelor's degree,... I was in the Air Force ROTC and you can't really get your commission and finally get out until you have a degree. So I didn't have a degree until I got a Master's degree and so then I got commission, but then the Air Force gave me a delay so that I could continue on for a PhD, which I did. So I ended up getting a PhD in 1958.

**Peter Asaro:** And that was also in electrical engineering?

**Nils Nilsson:** That was in electrical engineering, but it was on this business of using game theory to work on radar reception problems.

**Peter Asaro:** Was cybernetics influential in your thinking?

**Nils Nilsson:** Yes. Well, I read Wiener's book on cybernetics, although I didn't understand all the mathematics of it at the time and I thought control mechanisms were interesting. As a matter of fact, these kinds of ideas about control and feedback played a role in some later work I did after I was-- after I finished being Department Chairman of Stanford. Maybe we'll get to that.

**Peter Asaro:** So then you wound up an SRI.

**Nils Nilsson:** No, I wound up first in the Air Force because here I was at Stanford in the ROTC. Joined the ROTC because the Korean War was going on at that time and if you were in an ROTC unit, you wouldn't get drafted. Finally after getting my PhD, I had to actually go in the Air Force for three years and I was at a place that was then called the Rome Air Development Center in Rome, New York. It was part of an Air Force base called Griffiss Air Force Base. I think later it became Rome Labs and I'm not so sure-- I'm not sure whether it even exists anymore, but here I was as a lieutenant, doing work on monitoring the various projects that Rome Air Development Center was supporting with researchers around the country. So one of the interesting things was that among the projects I had to monitor were projects headed by people whose papers I read as a student at Stanford. So here they were, big wheels of the field, and I, a green second lieutenant, going to see if their work was up to standard. <laughs>

**Peter Asaro:** What were some of those projects?

**Nils Nilsson:** Well, they all involved radar work and there was a special technique that was being explored in radar signal processing those days and it was called pulse compression. The idea was this: that the amount of energy that you can get back from a radar reflection depended

upon basically the length of the pulse that you could send out, but the problem with having a pulse of a certain long length was that then it was a little more difficult to tell exactly where the airplane was. You'd want a very short pulse for that. More importantly, it would be difficult to separate two objects, so if the pulses, from their echoes, overlapped, which they would happen to do, it'd be more likely to overlap if they were long pulses and if the objects were close, then you wouldn't be able to distinguish there were really two objects. But if you had very short pulses, they would tend not to overlap, you'd be able to tell that there were two objects. So the question is how do you get a lot of energy into a pulse and also you've got good resolution in space of the distance. So the thing about the short pulse that makes it good for a resolution was that a short pulse has very high bandwidth, so then there were people who said, "Well, we can get high bandwidth into a long pulse if we just sweep the frequency, the carrier frequency, of the pulse as the pulse-- throughout the length of the pulse and then we can invert that process in the receive and make those pulses short again." And so these things were called matched filters. <coughs> Excuse me and that was most of the work that I was monitoring at that time. People were interested in very short pulses because they wanted to be able to detect different objects in the re-entrant ballistic missiles. So, for example, if an enemy were sending a ballistic missile in and it split into several different components, you'd want to be able to pick up all of them, so you want very high bandwidth pulses, which you could get with these long frequency modulated pulses.

**Peter Asaro:** Did any of that work end up being helpful for thinking about robotics?

**Nils Nilsson:** I don't think so. <laughs> I don't think so.

**Peter Asaro:** So were there any other interesting projects in the three years of the Air Force?

**Nils Nilsson:** Well, there was very-- a lot of interesting pattern recognition in terms of being able to tell what kind of radar you might be receiving signals from and that work on pattern recognition actually used some of the ideas that I had explored and studied during my thesis work on statistical detection theory. And so one of the things that happened in pattern recognition and increasingly so in the later years of artificial intelligence research is just how important statistical methods were-- various Bayesian techniques. And so that background that I had mainly applied to radar in my days when I was in the Air Force came back to help me later when I was doing work on pattern recognition.

**Peter Asaro:** So how did you wind up an SRI?

**Nils Nilsson:** Well, I was looking around. I knew I wanted to be on the west coast and I'd earlier accepted a job before going to the Rome Air Development Center at the RAND Cooperation. The RAND Cooperation thought, "Well, we can probably get you stationed here in Santa Monica

for your Air Force tour.” As it happened, they didn’t and I was assigned to Rome, so-- but I came back, I talked to the RAND Cooperation again and I talked to SRI and also to Stanford. SRI made me a nice offer and it was more exciting because although when I went to SRI, I thought, “Gee, it would be interesting if we could get the SRI people to do research on radars.” I talked to a very charismatic person at SRI who turned out to be my boss, Charles Rosen-- Charlie Rosen -- and he said, “Radar, that’s kind of like light bulbs. I mean, isn’t that a solved problem? Let me show you something very interesting that we’re doing here in the back room. We’re building a giant brain.” I went, “Oh, really? Is that right?” Well, they were building a very early version of a neural network and they had done some consulting with a professor from Cornell University, Frank Rosenblatt, who was one of the founders of some of the neural network research and so I met Frank and I ended up going to work for SRI in Charlie Rosen’s apartment doing work on these neural networks.

**Peter Asaro:** So was it a direct copy of the perceptron at Cornell or was it a variation?

**Nils Nilsson:** Well, it was a variation in the following way: mainly in details. Most of it in terms of its design-- schematic design was a direct copy, but the details were different. I think Frank Rosenblatt in his very early perceptron, to make the weights adjustable-- these weights are important connection units in the neural-- in the units that simulated-- so called simulated neuron, those weights were adjustable by having potentiometers motor driven, so he could send a current to the motors that would then change the potentiometer setting and thereby adjust the weights. We had, at SRI, a somewhat different technique developed by a physicist named Ted Brain and that was kind of interesting also-- brain. That involved magnetic—multiaperture magnetic core, so the amount of weight was stored as the amount of flux in this magnetic core and it could be adjusted and it could be read out.

**Peter Asaro:** Were you at all aware of Ashby’s work on homeostats?

**Nils Nilsson:** Yes, knew about Ashby, knew about homeostats, knew about some other people in Britain. <phone rings> I think we’ll just let that go. Well, maybe I ought to get it if you don’t mind because it might be a step son. Hello? Where were we? Oh, we had these magnetic aperture.

**Peter Asaro:** You were talking about some of the British guys.

**Nils Nilsson:** Oh, the other British guys. Oh, there was an Italian guy named Gamba, G-A-M-B-A, who had some kind of perceptron-like device. Maybe some other British names will come up.

**Peter Asaro:** Well, there were Grey Walter’s tortoises.

**Nils Nilsson:** Grey Walter's tortoises. However at that time, in the earliest days of my work at SRI, we weren't too interested in robots. We were building a pattern recognizing device, something that-- well, the Army Signal Corps was supporting the work. Now I think it's called the U.S. Army Electronics Command, something like that. The Signal Corps was supporting it and their rationale-- we didn't particularly care what the rationale was, we just wanted to do pattern recognition work. The rationale was to recognize symbols that were maybe hand printed on army maps. We actually ended up doing quite a bit of work on hand printed FORTRAN coding sheets, so we wanted to recognize-- in those days, computer programs were written out on coding sheets and then key punch operators read the coding sheets and punched what they saw on the coding sheets onto punch cards. So we wanted to eliminate that step by having that done automatically and so we actually got pretty good results and one of the reasons we were able to get good results was that not only did we try to recognize each character, but before we would accept a line of FORTRAN code as recognized, it would have to be something that could be compiled. So we had access to the FORTRAN compiler and we had to have a legal FORTRAN statement before we'd claim it was recognized. And to get a legal FORTRAN statement, we might have to go back and accept maybe the second best guess on an individual character. We had a ranking when we recognized each character as-- well, it might be ninety-five percent an A or might be eighty percent a B and so on and so forth and so we would go down the line until we could get the most probable, going across the whole sequence of characters, recognition of the whole line.

**Peter Asaro:** Did you come across the work of Hans von Forester on the biological computer line?

**Nils Nilsson:** Yeah. Now he was at the University of Illinois. As a matter of fact, I think when we came across his work, he might have been retired at the time and living on the west coast in Half Moon Bay. There was this whole field of cybernetics that had sprung up after Wiener and a lot of it was kind of woo-woo you might say. There are all kinds of strange ideas-- self-organizing systems. There were conferences on self-organizing systems and on cybernetics. Some of it was extremely speculative and not terribly well grounded, but that didn't bother us too much because what we were doing is pretty speculative also. A lot of people didn't believe in neural networks at the time.

**Peter Asaro:** Did you go to any of the self-organizing conferences?

**Nils Nilsson:** I went to some of those conferences, as did Charlie Rosen. As a matter of fact, at one of them, I think it was in Las Vegas, I met Warren McCulloch and Seymour Papert and McCulloch had all kinds of interesting ideas about perceptrons and how they ought to be built. A lot of those conferences, Frank Rosenblatt attended and Marvin Minsky. They used to have

fierce arguments. I think Minsky thought that perceptrons probably were quite suspect, not very scientific and...

**Peter Asaro:** Well, he published that book at the time as well.

**Nils Nilsson:** There's a lot of misunderstanding about that book I think, just in case. Some people think that the result of that book was a diminution in the research on perceptrons and it was even felt that Frank Rosenblatt quit his work on perceptrons after Minsky and Papert published that book and I don't think that was true. Actually Rosenblatt got very interested in something quite different from perceptrons. He got interested in transfer of memory in flatworms and so he would train flatworms to go to a maze where they would turn left or turn right and I guess they could be trained to do that, and they would more often go left than right if that's how they were trained. And then he'd grind them up and extract some polypeptides that he thought were some kind of marker of memory, feed them to naïve worms and then check to see whether these naïve worms, without any training, would go off-- go left more often than right. Well, he had a slight tendency, but most people thought it wasn't statistically significant and so he was doing that work, unaware that Minsky was writing this perceptron book.

**Peter Asaro:** What was Rosenblatt like to work with?

**Nils Nilsson:** He was a very smart guy and very intuitive. Although he had a lot of mathematics in his book called Principles of Neurodynamics, a lot of the mathematics was, I don't know, maybe not quite as rigorous as it might have been. Very early, he had this scheme for how to train a perceptron and it was thought and he thought he had proved that if you could train a perceptron to distinguish-- dichotomize, say, between some characters that this training procedure of his, which you call the error correction procedure, would lead to a correct dichotomization. His proof wasn't too good and a colleague of his, H. David Block, who was also a very smart guy at Cornell, ended up with a proof, as did some colleagues of mine at SRI on that. Matter of fact, it isn't difficult to know that there is a procedure which would lead to a correct set of weights that would lead to correct classification if it could be done. One could just try all possible values of weights until you got one. This was a little more direct, this particular error correction procedure.

**Peter Asaro:** What was Charlie Rosen like to work with?

**Nils Nilsson:** Charlie Rosen was extremely intuitive and quite bright. He was willing to try all kinds of things. I remember Charlie Rosen looking at various ways for controlling or for adapting multiple layers of weights. The perceptrons at the time only adjusted the weights in one of the so called layers between neural elements and we thought that, as did a lot of other people,

that if you had perceptrons with several layers, that they would be more powerful. But the key problem was how do you adjust all the layers, and so Charlie had various schemes he was cooking up, as did we all, and he would sit with quadrille paper, hand-simulating different kinds of pattern recognition problems, seeing if his schemes for adjusting multiple layers of weights might work. Actually later, David Rumelhart and others, probably working on some work that someone named Paul Werble [ph?] set down earlier, had come up with a-- essentially a steepest descent method for changing weights and we didn't do it because we didn't think you could do steepest descent backward through a threshold element. And their trick was well, you don't necessarily have to have a threshold element, just have some differentiable function that looks sort of threshold-like, the so called Sigmoid or S curve and you could back differentiate through that and that's what Rumelhart and colleagues did and they came up with methods, although they were quite slow for being able to adjust the weights in several layers, which did in fact lead to more powerful machines. Actually, if the first layer doesn't properly separate or partition, the inputs of different categories, no work by future layers is going to be able to partition them. The future layers, all they can do is then properly aggregate different parts of the partition space, so they'll go into the same category, so-- but the first layer has to do its job.

**Peter Asaro:** When you got to SRI, was it a large group working in this area?

**Nils Nilsson:** When I got there, there were probably eight or nine people in a little group called The Learning Machines Group. The laboratory that Charlie Rosen ran-- the laboratory itself was called The Applied Physics Laboratory and the other group in the laboratory was working on something quite different, but something that actually motivated Charlie to get interested in neural networks. The other part of the laboratory was headed by someone named Ken Shoulders. Ken Shoulders had this idea that if you could shrink at triode down to microscopic size, the field effect would diminish-- would eliminate the need for a heater on a cathode. So you could just get such intense fields with reasonable voltages across very, very short distances and then electrons would just boil off the cathode on their own and you could stick a grid in there, so he was interested in micro machining techniques for making thousands and millions of triodes. Charlie thought, "Well, it's going to be difficult to wire all these things up. Maybe we can develop some sort of self-organizing idea." He had gone to some of these self-organizing conferences. That would allow us to get them to do something useful, so that's when he started talking to Frank Rosenblatt, before I came to SRI and got interested in neural networks. Originally his idea was that they would be the-- that would be the technique that would wire up these triodes. Of course, the transistor came along, making it-- making tiny triodes not the best way to go. Now we can do billions of transistors and they don't have any problem wiring them up. You just use a photo mask and lay down the wires wherever you think they ought to go to do the logic circuits and so on.

**Peter Asaro:** So who were some of the other researchers that were working in your lab at the time?



**Nils Nilsson:** There was a man named George Forsen. He was primarily the engineer involved in building <inaudible>. There was, as I mentioned, Ted Brain, who was originally from England and it was Ted who told us about Uttley and some of the others who had done work in England. Soon after I came, we hired a couple of very smart additional people. One was named Peter Hart. Peter Hart eventually became my boss at SRI because I wasn't too interested in management and he went on to found a laboratory at Schlumberger, a company, and also more recently was chief of a laboratory that Ricoh started in Menlo Park, Ricoh the camera company. And we also hired a man named Richard Duda, Dick Duda, and Dick Duda was a very bright mathematically included person who made sure of the mathematics of all the things that we were doing that involved work, even in the neural network days, through some expert system work that we did, was well founded and he's published a lot of papers.

**Peter Asaro:** They had a textbook and pattern recognition?

**Nils Nilsson:** Duda and Hart. Now the guy that Peter Hart hired at Ricoh, David Stork, wrote the second edition-- Duda, Hart and Stork, although Stork did the writing.

**Peter Asaro:** So at what point did the idea of making a robot start to take shape and how did that take shape?

**Nils Nilsson:** Well, let's see, first of all, Charlie attended a class on programming and Lisp. We knew through going to conferences and being close to Stanford that there was this subject called artificial intelligence that we vaguely thought neural networks probably were part of and pattern recognition was part of, but what's it all about? Well, Charlie thought we ought to find out, so he went to a class given by a recent graduate from MIT, a PhD student from MIT, a student of Marvin Minsky's named Bert Raphael. Bert Raphael taught, of course, on Lisp. Charlie came back all excited thinking, "We've got to do something about A.I. We've got to write programs and there's all this stuff going on and we should be part of it." And later, by the way, we ended up hiring Bert Raphael. He had a job for a while at UC Berkeley, but then joined our group and so Charlie thought, "Well, we have this pattern recognition work. That could account for the robots perception, especially vision" and through Lisp programming and through various types of heuristic programming ideas, we could probably build control programs for a robot, which Charlie called an automaton. And he said, "We need to build this automaton" and it would roll around and it would recognize its environment and we could get it to do things. Then he wrote a memo and I haven't been able to find a copy of that memo, but the original memo got all of us excited and was used as kind of a fundraising device to try to get sponsors interested in this project. And that was probably in around '63 or '64 and finally-- a lot of this is detailed in-- some detailed in my book on the history of A.I. We did get our project supported by the Advanced Research Projects Agency called ARPA at the time. Now it's been renamed DARPA-- for several hundred thousand dollars over some year to build this device that would roll around

and we had engineers who were reasonably good, but maybe not as good as they could have been at actually building the motorized hardware in the vehicle because they might not have been as good as they could have been. When it started or stopped, it shook and so we ended up calling it Shakey. We had a session to talk about how we were going to name this machine and somebody suggested Shakey. Then a very interesting person, whose name I might reveal, said, "Absolutely not. This is too important a project to give it such a frivolous name," so of course we did give it that name. <laughs>

**Peter Asaro:** So who was that?

**Nils Nilsson:** That was a guy named Stephen Coles. You might want to talk to him also. He's got a movie out, which I think I have a copy of online, of the robot history-- little clips from JPL and a lot of other places. Well Stephen Coles participated in the project. He was a student of Herb Simon's at Carnegie Mellon and he joined us at SRI. He was interested in natural language processing and so his idea was we want to be able to talk to Shakey, actually not through speech, but by typing, and he wrote a little grammar and a phrase recognizer and so on that would be able to recognize certain symbol commands that we gave it, so he did work on that. Then later, he decided he wanted to go to medical school and he ended up going to the Sorbonne, got a degree, a French medical degree, and I think most recently he's at an institute of gerontology at UCLA and he's available. I mean, I have his email address.

**Peter Asaro:** Okay, yeah, I'll track him down. So what were some of the challenges?

**Nils Nilsson:** So let's see, there's other people too involved. So anyway, we built this thing and it moved around and we made a film of it that's online. I think they had sections of it at the computer history museum and it was the first controlled robot device, although there had-- it was preceded by a similar device at Johns Hopkins University, but I don't think it was quite as sophisticated and they called it The Beast.

**Peter Asaro:** It was more of an analog system.

**Nils Nilsson:** Yeah and of course, all of Grey Walter's things. So what it could do, it could make a map of its world, it could use a map, it could up-- well, it didn't really make the map entirely. It could use a map and it could update its own position in that map and it could locate objects in the map. If there was a new object, it could create it and put it at the right place in the world and so we wanted it to be able to solve certain kinds of problems. We had two innovative ideas at the time that we wanted it to be able to capitalize on. One was navigation, so how do you get from A to B and so there were various techniques that we explored, some of which involved-- most of which-- all of which involved graph searching. So if you could represent the points--

possible points in the space that you're going through as points in a graph and the legs and how you would get from one point to another as branches of that graph, you could search the graph for the shortest path and there were various techniques for graph searching that were around at the time. Ed Jordestra, [ph?] for example, had one of them and so we came up with this technique of-- well, there was another graph searching technique worked on by a team at the University of Edinburgh under Donald Michie, called the graph traverser, and they used heuristic estimates essentially for deciding which place you ought to look for first as a way-- as a possible path. And we modified that a bit by saying, "Well, you could score the different places that you ought to look at next by what we called a value function and the value function should consist of two parts we thought. One part was how long was the path from the start to the place you were considering? You had gone so far and the next part was how far the goal was, which you guess of how far the goal was from there, and well, you don't know how far the goal was until you actually complete the path, but you could make an estimate based on what we thought the airline distance would be, so the Euclidean distance of the path. So we used that as the heuristic function, we called it, and used that in an algorithm that we called A star. Star was a little asterisk after the A. Intellectual engineering very often optimal methods had that little asterisk after them-- optimal estimates. So we said, "Okay, we'll call it A star" and we were able to prove that if the heuristic estimate was not any greater than what the actual path length would be that you would soon discover, then you could prove that that particular method would in fact find the shortest path, as soon as it found the goal and it turned out the goal was the next place that you should look at, that that would be the shortest path. And you could generalize it to least costly paths. It doesn't have to be distance. Any kind of cost function could be put on it. It turned out that that particular method is currently used in all kinds of applications and it is the basis for some of the map searching applications at Google and MapQuest and some of the others use. Unfortunately, you couldn't patent things in those days and so we don't get a tenth of a cent for every time anybody uses it.

**Peter Asaro:** Too bad. I used that to get here.

**Nils Nilsson:** Probably, yeah. Did it work? <laughs>

**Peter Asaro:** Yeah, I'm here.

**Nils Nilsson:** So that was one thing. The next thing was, "Well, we're interested not only in navigating through a space-- a geometric space, the laboratory that we had, but we're also interested in navigating through this other conceptual space." If you're trying to achieve a certain goal, which could be stated in a special language. We used Logic. And you had certain actions that you would perform that were advertised to achieve certain subsidiary goals. How could you string these actions together into a path which would take you from your present position to achieving the goal, not necessarily the goal of being at a certain place, but the goal of having a

certain combination of things satisfied? And that led to a program that we call STRIPS, which was an acronym for Stanford Research Institute Problem Solver and so it was used as the main problem solving device. So one of the key developments, I think out of the-- besides the A star and the STRIPS algorithm, we had a lot of other algorithms for vision-- processing scenes and doing line finding, although other people were working on these problems also at Stanford and at MIT. But one of the things besides STRIPS and A star and the vision things was putting together the architecture in what we call the three level layer architecture, which is currently used in a lot of robotic systems today. At the bottom level were some of the very basic motor functions, which really sort of programmed into the robot. You didn't have a lot of control over how they were going to be modified. They were the basic things a robot could do-- move forward, move left, tilt the camera, pan the camera. And then we had, in addition to that, at one level higher, a level that we call the intermediate level actions and the intermediate level actions were ways of stringing those basic motor functions together and we used a special technique called Markov algorithms, not too unlike what Rod Brooks and others later developed in their so called subsumption architecture, for putting those things together. They were quite robust in the sense that it involved this whole idea of feedback that I was struck with from knowing about cybernetics. That particular intermediate level action didn't finish until it achieved its goal even if it had to go backwards and start over and do some things over, so they were very robust in the sense that they were stubborn about getting to the goal and if there was some slip up, "Well, okay, well now where are we and what's the best thing to do from here?" So then the highest level was where STRIPS operated. The highest level would then string together these intermediate level actions, assuming that they were robust and that they actually would work, which they often did-- string them together into a plan to achieve a goal. In that case, in goals were mainly things like pushing boxes from one place to another, going through-- getting from one room to another, going through doorways, unblocking doorways if you couldn't get through, things of that sort, so we made a movie of all of that sort of thing happening.

**Peter Asaro:** How long did it take you to film all of those different things?

**Nils Nilsson:** Days, <laughs> which we then were able to sequence into a reasonable movie. The movie itself-- what we did is we ran the program and got all the results, but we ran each section as kind of a sequence and then we would start it up again because we'd have to move the camera position, things of that sort.

**Peter Asaro:** Did you have to reprogram the robot each time as well?

**Nils Nilsson:** Well, I think probably in the early days of getting ready to make the movie, we were always introducing new ideas into the program, maybe making STRIPS more efficient, but we didn't have to do any reprogramming actually during the movie making. It was all what we had and frozen at that time.

**Peter Asaro:** Were there any outtakes?

**Nils Nilsson:** Yeah, there probably were things. I don't recall at the moment what they might be, but there might have been glitches, maybe computer crashes or things of that sort.

**Peter Asaro:** And it has these wires around the base. Those are collision detectors?

**Nils Nilsson:** Well, Shakey's system had several sensors. It had the television camera. It had a triangulating range finder using a laser, not a time of flight laser, but a triangulation system and it had these so called cat whickers that would trigger switches if it bumped into something. It also had a bump bar for pushing things and when it was engaged, that is to say you were really up against a block, and the bump bar would get engaged and you would know that you're actually in contact.

**Peter Asaro:** And after that initial set of experiments, did you continue development on Shakey?

**Nils Nilsson:** The movie, I think, was taken in about 1971 or 2 and that was probably pretty much the end of the Shakey project. We had a lot of ideas about how we wanted to expand the work, things which Shakey couldn't do. For example, Shakey didn't know anything about what other actors might be around and couldn't coordinate with other actors. As far as Shakey was concerned, there was no dynamics in the world. The world stood still. The only time anything changed is when Shakey changed it. So there were things of that sort we wanted to do and we went to-- we wrote proposals, we went to ARPA and ARPA, at that time-- well first, ARPA had a succession of different managers and the manager at that time decided that ARPA wasn't interested in any more robotics work and so for a while, we tried to do the work we wanted to do under a different label. We developed what we called the computer based consultant, which was going to be a system that would advise a person, an apprentice, how to accomplish various tasks. So in a way, the apprentice was the robot. We still had a problem solving system above all of that, which would sense the environment for the apprentice and make plans for how the apprentice-- one of the tasks we considered, for example, was taking apart and reassembling an air compressor. So we would start out with the air compressor assembled and we would tell the apprentice each step about what to do and then how to put it back together again and so the planning system would then figure out all these steps. But then ARPA finally decided no, it didn't want to do that either, so we had to leak off into other things.

**Peter Asaro:** Why do you think ARPA lost interest in robotics?

**Nils Nilsson:** Well, a lot of times-- I think it's still the case with ARPA-- there are certain fashionable topics that either through their perception of what congress wants or their perception about what the military demands are motivate ARPA to work in certain subject areas. I think around that time ARPA got very interested in what they call the command control. So given a situation on, say, a battlefield, when you're getting all kinds of information from different sorts of sources, how could a commander put all of that together and make a decision as to what the current status is and what should be done? And so there were various tools for command and control-- tools that were thought to be helpful to a command and control-- mainly tools-- one of them that SRI got involved in is getting better access to databases and more convenient access to databases. Could you do it through a natural language system? Could you ask questions of the database and get answers? More than that, the information that you might want might actually reside in several databases, so could you have a system that would decide how to parse out the particular question, getting this part of the answer from database A, another part of the answer from database B, then going out to the databases, getting those answers, then putting them back altogether to give an answer to the commander, say. So SRI developed a rather large project doing work in that area.

**Peter Asaro:** What was that project called?

**Nils Nilsson:** Well, there was a project called TEAM, transportable English access to something. I think I have all those acronyms in my A.I. history book, which we can look up and there's a good description of that work involving natural language access to databases that I talk about in the book.

**Peter Asaro:** So during that time that you were working on Shakey, was there much interaction with John McCarthy or any of the people at the lab at Stanford?

**Nils Nilsson:** Yes, I think there was quite a bit. As a matter of fact, in-- yeah, in 1968, I spent kind of a half a year on loan from SRI to Stanford and I had an office out at the Artificial Intelligence Laboratory that Stanford was running on Arastradero Road, in the foothills behind Stanford and worked with-- there was other people working on-- mainly on machine vision and other robotics projects. One particular person I remember quite well is a man named Robert Bolles who was working on automatically assembling a water pump for an automobile and then they were doing various puzzles. Could they solve the instant insanity problem, as they called it, by putting blocks together? And so yeah, there was quite a bit of interaction.

**Peter Asaro:** And he wound up eventually at SRI.

**Nils Nilsson:** He did. He's there now, I think.

**Peter Asaro:** And still working on vision.

**Nils Nilsson:** And Jerry Feldman was doing work on robotics and vision in those days. He later became the founding chair, I think, of the computer science department at Rochester and now he's at Berkeley in an institute called The International Computer Science Institute.

**Peter Asaro:** When did Ruzena Bajcsy show up?

**Nils Nilsson:** She got her PhD at Stanford and let's see, where did she go? Did she go immediately to Penn after that? I know she spent a little time as an NSF, National Science Foundation, director and-- but she did work at the A.I. lab.

**Peter Asaro:** And did you run into her there?

**Nils Nilsson:** Oh, yeah.

**Peter Asaro:** And what would you say was the kind of intellectual discourse that was going on of the exchange of ideas?

**Nils Nilsson:** Well, I think it was very exciting. That whole period of the Stanford A.I. lab when they were in the foothills was a very exciting time and they developed a lot of things-- probably the first printer that led to the-- eventually to the Xerox printer. A guy named Les Earnest was John McCarthy's Chief Executive Officer, you might call him, and he was interested in a lot of different things. They had a lot of extra computer cycles out there and so they allowed people interested in computer of music to use it. John Chowning was out there at the time. John Chowning, by the way, was a person who invented the frequency modulation technique for sound synthesizer that led to a lot of money for Stanford, paid for by Yamaha, who included it in their keyboard stuff. And then eventually, they got a site on the campus called the Center for the Study of Computers in Music or something like that. There were a lot of things going on, not all of which were necessarily connected to A.I. After all, you had this very powerful computer-- time shared computer-- PDP-6 and eventually PDP-10s and people understood that you could do all kinds of things with them. You could do text processing. You could do all kinds of pattern recognition things. Ed Feigenbaum was there for a while working on the project which became Dendral, a program for deciding about the structure of small organic compounds. Ultimately he left the A.I. lab because I guess he couldn't get enough cycles on the machine and moved to campus. Well, he was upset because McCarthy and his people were always changing the machine around and taking it down and making changes to it.

**Peter Asaro:** Did you ever work on another robotics project after Shakey?

**Nils Nilsson:** In my life?

**Peter Asaro:** Yes.

**Nils Nilsson:** Well, in a way. Maybe we'll get to that. If you want to go sequentially, just briefly, I was a little bit miffed that ARPA wouldn't support robotics and I decided that I wasn't going to have anything to do with what ARPA thought they ought-- that I ought to get involved in. I mean, I knew what I wanted to do and so I left ARPA's support and together with Peter Hart and Dick Duda, I worked on this so called prospector expert system. We were motivated by work by Ted Shortliffe, a PhD student at Stanford, who had developed an expert system called MYCIN, which is sort of the suffix of some medical drugs, Romycin and things like that, Streptomycin. So anyway, he had some ideas for putting the knowledge of experts together in a way which was usable and could come up with diagnoses and so Peter got the idea-- or maybe it was Peter and Dick. I'm not sure who actually thought of it first-- that perhaps you could use the expertise of geologists, explorational geologists, to help decide about where mineral deposits might be. And so they talked to a lot of people at Stanford in the Geology Department, students and faculty, and put together an expert system that they call Prospector, which underwent some development and it ended up accurately predicting a mineral deposit in the city of Washington at Mount Tolman, a molybdenum deposit. And actually it wasn't at the time, it couldn't be called an ore deposit because in economic geology, an ore is defined to be something that's economically profitable and at the time, I think the price of molybdenum was not high enough, but it was a mineral deposit and it was reported in Science magazine. So I was doing work on that. Then pretty soon, Peter Hart decided to leave and start a company and they needed a new person to be in charge of the A.I. laboratory. I had, up to that time, tried to resist management types of things, but I was the obvious candidate I guess, so I did it. That was, I think, in 1980 and so I was in charge of the A.I. lab, which I think we called an A.I. center, until I went to Stanford as Chairman of the Department in '85. So at that time, during those years, I wasn't doing a lot of my own research because I was doing mainly management things.

**Peter Asaro:** How did they corral you into being the department head of Stanford?

**Nils Nilsson:** Well, if I didn't do it, who was going to? I mean, so it was a matter of, "Okay, so you've got to take your turn." That was at SRI and then how'd they corral me...

**Peter Asaro:** At Stanford.



**Nils Nilsson:** How'd they corral me into that? Well, I had taught courses at Stanford in the past and I had students who had worked with me at SRI while they were doing their dissertations at Stanford and so I had a pretty good connection with Stanford and I thought, "Well, maybe I'd like to actually go over there and be a faculty member," but they weren't, at that time, hiring any new faculty members, but they were looking for a Chair and so I thought, "Well, that's one way to get over there," so I took it on. I promised I'd do it for six years and did.

**Peter Asaro:** I want to get back to your students, but first I want to ask when you stopped getting ARPA funding, what other places did you go to look for funding?

**Nils Nilsson:** We had funding from the Office of Naval Research. I was writing some books at the time throughout the seventies and even into the eighties and the Office of Naval Research supported some of that. We had projects from the Air Force Office of Scientific Research. The work on the Prospector expert system was largely supported by the U.S. Geological Survey. We had some support from NASA, so we cobbled together projects here and there to support the kinds of things we were really wanting to do.

**Peter Asaro:** So who were some of the PhD students who supervised in working on Shaky?

**Nils Nilsson:** Well let's see, yes there were quite a few. We had Tom Garvey who's now at Stanford-- at SRI still, I think. We had a young man named Earl Sacerdoti. Not on Shakey, but later, Kurt Konolige, who was at SRI for a long time. Got his PhD at Stanford, worked with me on a system for reasoning about knowledge and belief and now he's actually at a place called Willow Garage. Have you seen Willow Garage?

**Peter Asaro:** Yes.

**Nils Nilsson:** You probably saw Kurt.

**Peter Asaro:** We didn't talk to Kurt.

**Nils Nilsson:** Actually, I hiked across the Sierras with Kurt on skis-- Kurt and a bunch of other young guys. I was probably in my late forties at the time, but we did a tran-Sierra trip in the winter, camped out in the snow for five nights, went from east to west over the Sierra crest. Let's see, we had Douglas Appelt, A-P-P-E-L-T, one of the students. We had a guy who did a lot of the Lisp programming for Shaky the robot named Robert Yates, Bobby Yates, Y-A-T-E-S. We had Cordell Green, probably one of the more instrumental of the people involved in robotics while I was at Stanford-- I mean, while I was at SRI. Cordell had read some papers by a guy

named John Alan Robinson on theorem proving and so early in the Shaky days, we thought that the main reasoning mechanism for Shaky should be a system that could manipulate expressions in symbolic logic and there have been other techniques for doing so, but Robinson came up with a technique that was particularly suitable for computers and Cordell Green read the papers, knew about it and decided he wanted to program up a system, which he did. I remember him telling me, "Nils, you've got to learn about resolution." I said, "Resolution, what's that?" And so anyway, I read various papers and got involved in that, which lead some people to claim that I was overly interested in logic as the main approach for dealing with artificial intelligence. I think of it as one aspect of intelligence. We, humans, are sometimes logical and we put together logical arguments and logic plays a role in many of the things we do. It's not the main role, maybe, but-- so anyway, we put together a system for reasoning and symbolic logic that used John Alan Robinson's resolution principle. As a matter of fact, he was a component of our STRIPS problem solving system also.

**Peter Asaro:** Where did Green wind up?

**Nils Nilsson:** He was also in the ROTC. He was in the Army for a while, but I think as a matter of fact, he served mainly his tour duty at ARPA and he was an assistant to Larry Roberts, who was the head of an ARPA office of information processing techniques office and it was Cordell Green who was assigned by Larry Roberts to look into whether or not it would be a good idea for ARPA to support research in speech understanding. It turned out it was a good idea and they did and that's all mentioned in my book also. So then Green started a company that is still in existence as far as I know.

**Nils Nilsson:** As a matter of fact, he served mainly his tour of duty at ARPA and he was an assistant to Larry Roberts who was the head of an ARPA office of information processing techniques office. And it was Cordell Green who was assigned by Larry Roberts to look into whether or not it would be a good idea for ARPA to support research in speech understanding. Turned out it was a good idea and they did. And that's all mentioned in my book also. So, then Green started a company that is still in existence as far as I know, what's the name of the company, it may come to me. But, it's a company in Palo Alto that did work on software engineering.

**Q:** Cordell.

**Nils Nilsson:** We had lots of students come from Stanford to SRI. And I learned a lot from the students because my background, I didn't know what a compiler was. I mean, I didn't have any background in computer science, and so these students would read stuff and tell me how to go learn it and I did.

**Q:** Did you have my much interaction with the people who were working on manipulation and grasping?

**Nils Nilsson:** Well, I knew about Vic Scheinman, I knew about his arm that he developed and designed at Stanford. But, I didn't personally have any interaction with him. I wasn't so interested in arms at the time, Shakey didn't have arms. Although we designed a system that we had proposed to ARPA that would have arms. But, they ended up not funding it.

**Q:** When did you make that proposal?

**Nils Nilsson:** Oh, I don't know, probably in the early seventies.

**Q:** And did you have much interaction with Bernie Roth?

**Nils Nilsson:** I knew who he was, but no professional interaction. I mean, we didn't work together on anything.

**Q:** Okay. And then, at what point did your work ever come back to robotics?

**Nils Nilsson:** Well, so I go over to Stanford and first thing is well, I had to learn about how you become a chairman. The atmosphere was a good deal different than it was at SRI in a number of ways. First of all, the people in the department don't actually work for you. I think their idea is look, you're in charge. We're passengers on your ship, we'll tell you where you want to go. Your job is to keep the engine running and do all the things that you need to do. And so, that was a bit of a change, but one that I didn't mind, had a lot of smart people at Stanford. There were occasional things we had to solve problems as any manager does. Sometimes there are things that happen on the tail of the distribution that you got to go deal with. The other thing is whereas at SRI you could work at night if you wanted, but largely you'd get there in the morning and you might leave at six o'clock at night and that was the working day. But, at Stanford really your work is never done. People are emailing you, they got problems some of them in the middle of the night. And they need things done plus the fact that there's this level above the department chair, the deanery, and deans have various demands.

I mean, they want a vision statement every now and again. Where's the department going? We have to have tenure cases. So and so is up for tenure, you have to make sure we get all kinds of letters written. And you have to present the case for this particular person's tenure in front of a committee. We need your budget and oh, by the way we're going to have a special committee in the university to look into how we can better use computer science in civil

engineering. Would you be on that committee? And so, there's all that kind of thing that happens more at a university, I think, than a place like SRI. So, it was a lot of work and the first thing I discovered there was quite a bit of work. But, it was all very exciting and challenging so I liked it. I did get involved right away with a bunch of students and I had four or five of them, Ph.D. students. One of the difficulties was that they all came in at the same time so in our little group we had weekly meetings. They didn't have anybody, any kind of mentors, people who had gone on before them. People they might work with, so they're all starting from scratch and that made it kind of difficult probably for them and for me, but they had some very good students. One of whom, by the way, did end up at SRI.

**Q:** Some of your Ph.D. students and one in particular.

**Nils Nilsson:** Yeah. One of them named Karen Myers went to SRI and played a leading role in the project at SRI called CALO which stands for cognitive assistant that learns and organizes. It was a multimillion dollar project. And it actually led to a start up company called Siri, S-I-R-I, Siri, which has an application, free app I think, you can get on your iPhone that will give you advice about where to go to dinner and making reservations for you, things like that.

**Q:** So, I mean, did you do any specifically robotic work after?

**Nils Nilsson:** Well, let's see. In terms of robots after finishing being chair I took a sabbatical and went to, had half a year at MIT and Harvard, some combination, and I was in Rod Brooks' lab and I would teach a class or two at Harvard once in a while, not a class, but I'd lecture for someone who was teaching a class. And half a year at Santa Fe Institute in Santa Fe, New Mexico and during that time one reason I wanted to go to Rod Brooks' lab is I began to be interested again in these intermediate level style programs which were very robust. And I was developing a formalism, a style of programming that I called teleo-reactive programs. And these teleo-reactive programs, I called them teleo-reactive because first of all the teleo part was they had a purpose. And the reactive part was how they worked toward that purpose depended upon their particular reaction to the current situation. So, however they decided to achieve the goal depended upon where they were at the moment. And if where they were at the moment suffered a setback as on their way to the goal something untoward would happen, well they would react to that and carry on.

So, I had some students at that time who were developing in software simulated robots which we called bots. Well, I did have one student who actually programmed a little NOMAD robot using this teleo-reactive formalism. Tom Willicke, yeah, I think his name was Tom Willicke. And another student named Scott Benson who used this particular formalism to program simulated devices that would learn the effects of their actions. So, he was able to develop a system that could learn to fly a simulated airplane, they have these flight simulating

programs. And so, his program could learn by watching and instructor how to take off and how to land. And it would learn that by noticing all the conditions that preceded a particular instructor's action, all the readings and all the dials and everything, and the conditions afterwards and then it would deduce or adduce that that particular action that the instructor performed would, in fact, take the situation from the before to the after. So, it would develop a whole set of these operators which were very much like the strips operators that I talked about earlier, the strips rules. So, he was able to do that. We never made a movie of it, but he wrote a dissertation and I think has had some impact.

**Q:** That's Ben?

**Nils Nilsson:** I wouldn't close that window again. We won't tell Grace.

**Q:** And what year was the sabbatical?

**Nils Nilsson:** You set, all your wires?

**Q:** It's just the headphones.

**Nils Nilsson:** That was in 1990. I went off in the fall, summer of 1990 and ended up in the summer of 1991. And so, I had another student who programmed little tiny robots that would wander around on a screen, these were simulated robots, using teleo-reactive programs that would accomplish various tasks. Another student who I'm still in touch with quite a bit was able to program some simulated soccer players using teleo-reactive programs. So, soccer player, although might make a plan for doing something, the plans don't always develop the way you think they're going to because there are other players, other things happen. And so, but you always got confronted with a certain situation and there's always some best thing to do in that situation. And so, these particular teleo-reactive programs would try to do that.

**Q:** Do you know if that strategy has ever been used for the Robo Cup competition?

**Nils Nilsson:** Not that I know of. However, I think that the idea of these kinds of reactive agents, I mean, I certainly wasn't the only one pursuing them. As I mentioned Rod Brooks was doing similar things. There was a man at JPL, Aaron Gat, G-A-T, who had some programming ideas that were quite similar. There was a man named Firby, F-I-R-B-Y, at University of Illinois, I think, was he? And I think he had something called reactive action packages.

**Q:** And would you say there's more similarity or more difference between say the subsumption approach of Brooks or some of these others to your own?

**Nils Nilsson:** Well, the subsumption architecture, I think, was quite a moving target. I wasn't too sure exactly what it was. It was anything Rod Brooks was thinking up at the time. I think the main thing that motivated Rod Brooks was that you could use the real world itself instead of a model. One didn't have to make models and work plans out in models. Let the world store itself. And so, most of his ideas on subsumption capitalized on that notion. To a certain extent we did also with the teleo-reactive programs although you could also make plans stringing together these teleo-reactive programs and then proceed from there. So, I think my idea was that yes, the world can store itself, but you people do use models. You have to remember where you parked your car if you're going to let the world store itself you'll eventually find it, but you got to look at a lot of places.

**Q:** So you've used both statistical and logical methods, models and representations as well as reactive systems. But these have often been very contentious debates within the AI community. How do you see that?

**Nils Nilsson:** Well, I think it's kind of working itself out now. I think that if one knows something for sure, I mean, statistical techniques, well, one way in which their used in these so called belief networks in which more or less you have propositions and propositions sometimes can be the cause of other propositions. And each one might not be known with certainty, you have various probabilities. And there are techniques for propagating probabilities throughout the network. So, if you learn something new that affects the probabilities not only of the new thing you've learned, but all kinds of other things that might be related. However, it is the case that sometimes you absolutely know pretty much with certainty or at least if it's 95 percent certainty you may as well take it as certain and then in that case, probably ordinary logical reasoning might be more appropriate say than trying to do things with probabilities. After all, when you have the probabilities if you're going to do everything exactly with probabilities there's a good deal of computational burden that not having certainty adds. But, if you do have certainty you ought to be able to capitalize on that extra knowledge. So, I think that there's a role for both techniques.

**Q:** Looking back what would you say were the most challenging problems in pattern recognition and what were some of the biggest breakthroughs that you had?

**Nils Nilsson:** Well, in pattern recognition one of the problems that was recognized from the very earliest days was separating the figure from ground. So, if you're looking at something well even today in face recognition you have to be able to isolate what the face is first and where it is before you can decide to apply the various techniques that you might be able to apply to decide

whose face it is. So, picking out a face from a crowd, I think that's largely been pretty well solved. I mean, everybody's personal computer if it has Picasa has a way to recognize various faces and isolate them. Cameras have face isolating mechanisms in them. But, I think it was thought to be a big problem in the past. Otherwise, in pattern recognition I think it's still the case that we're not very good at dealing with cursive script. Connected speech, well, there are various statistical techniques that do a pretty good job with it now. You can buy commercial products but being able to deal with connected speech and connected writing was an intermediate term problem that I think has largely been solved. You talk to people nowadays who are prone to use the various statistical methods and I think they say most of these problems that we thought were really difficult can be solved just by having lots and lots of data. The more data you have the easier it is to solve it and so many of the natural language understanding systems and the, for example, translation programs basically use just lots and lots of examples to help them.

**Q:** And what do you see as the big problems in the near future for AI?

**Nils Nilsson:** For AI? For AI in general. Well, dealing with common sense I think is still the big problem. The expert systems are pretty good in their special domains. You might say "Well, let's take a look at one of the milestones achievements." There's been several milestone achievements in AI. Chess playing was one, the autonomous car is another. More recently I think the "Jeopardy" Watson, the Watson program of IBM that competed on "Jeopardy" is pretty much a milestone achievement. However, I think even its designers will admit that when it came to dealing with common sense there were difficulties. It could get around that for certain classes of questions it was asked. The more circumscribed the question and the more likely it was that that question was explicitly in its huge database the better off it would be. However, it made several mistakes and even though it won and I think that a lot of the mistakes would have been solved if it had a better, more encyclopedic, common sense knowledge. There have been various attempts to try to solve that problem. Doug Lenat at Cyc is one who thought that "Well, the way to deal with it is to hand code all the different elements of common sense you might ever need," the hundreds of millions of little bits of common sense information. Well, good luck. That's a never ending task because common sense changes and culture changes and so you're going to have to have something that can keep track of it and how do you keep track of it? Well, the internet's a good source. There's just an awful lot of stuff on the internet. It's all in English. So, you have a sort of a chicken and egg problem to recognize something in English you have to have common sense, but in order to have common sense you have to be able to recognize all the things that are there. But, I think gradually we're making progress on that.

**Q:** And what about more specifically in the area of planning? So, historically would have been the big problems that have been solved and then looking to the future what are the big problems ahead.

**Nils Nilsson:** I think planning's pretty well under control now. And I haven't been keeping track of that literature, frankly. But, there are various programs that win so called ARPA planning contests. And they put together plans that seem to be quite beyond what humans would ordinarily do. Planning movement of supplies, doing things like per charts and for a long time I think the problem of dealing with plans which had a big aspect of time involved in them. Certain things would take a certain amount of time, these things could be done in parallel, these other things couldn't be done, how do certain time dependent processes interact? But, I think they've made a lot of progress in that although I haven't been keeping up with it.

**Q:** And in terms of so this project you did right after Shakey which was a sort of a coaching.

**Nils Nilsson:** Coaching, computer based consultant as we called it.

**Q:** This was sort of a form of human robot interaction, do you see areas in which AI will advance the ability of robots to interact directly with people?

**Nils Nilsson:** Well, there are a lot of people focusing on interaction between humans and robots particularly the social interaction amongst them. I think some people are, Cynthia Brazil at MIT and some others, haven't kept up with that either. But, yeah, I mean, humans are going to have to, well, we hope we don't put too much of the burden on humans, that they have to be specially trained to be able to interact with robots, but robots are going to have to understand humans and so, there's the need for, I believe, robots to have models of what humans believe, what humans' goals are, what it is people are trying to do. And so, for a long time there was work on what the psychologists might call theories of mind. How is it that one agent can understand what another agent believes and what another agent might conclude? And there were special logics designed for dealing with that. I'm not sure where that's going nowadays, but I think it's an important area for work. And robots will have to have some idea of what it is that the people around them are wanting to do, what their goals are, how the robots can either get out of the way or help achieve those goals.

**Q:** And you mentioned before that a lot of the methods that are successful are the ones that need lots and lots of examples. Do you think that's going to be a problem for robots that have to interact with the world, that some of their examples might be crashing into things and things like that?

**Nils Nilsson:** Well, there's no reason why you can't give them lots of examples. I mean, they don't have to in their life have experienced all those examples. You can just put them in which you can't do with humans. So, with humans you actually have to learn by living. But, with robots you can put the life of previous robots right into them.



**Q:** So, in the case of Watson were there specific things you noticed in the kinds of errors that it made that made you sort of think about how it was structuring common sense?

**Nils Nilsson:** I recorded all of that and I looked at it. I've forgotten some of the questions that were asked. Well, for example, on this Toronto thing. Remember the example in which it was asked, well the clue was World War II aviators or naval battles for which airports were named. And the answer would be Midway and Chicago and O'Hare. And it mentioned something in Toronto, I think, even though it was supposed to be a US city. And I think the people who designed it explained that well, oftentimes the category US city doesn't really mean exactly US city, it's sort of general. And so, Watson didn't count that as heavily as it should have. But, maybe it had some inaccurate common sense that allowed it to answer Toronto. But, anyway it was a failure of some sort of common sense reasoning.

**Q:** And do you think the increased power of computing is going to really advance artificial intelligence and robotics at a much faster rate?

**Nils Nilsson:** Well, if you had asked me that a few years ago I might have said "No," because I think that whatever the power of computing was at the time it was fully able to handle any of the ideas that we had at the time. I mean, we were idea short, we weren't hardware short mainly. Now, I'm not so sure because the new idea that's come up, the use of lots and lots of data, well, lots and lots of data requires lots and lots of computing. And so, the more computing, the faster it can go the better. I don't know that that's the bottleneck at the moment. After all if you talk to the Watson people which I haven't, but if you did talk to them and you asked them, "Gee, how could Watson have been better? If you had a computer, this IBM 7000 series or whatever it was, if it were 10 times as fast, 100 times as fast, 100 times as much memory would you have done better?" I think they'd still answer "Well, not necessarily. We would need more ideas about how to program all that."

**Q:** So, it's also about the structuring of the knowledge. Do you think there's specific problems in AI that aren't going to be susceptible to large corpuses of data or that require new understandings?

**Nils Nilsson:** Well, large corpuses of data are going to be useful in lots of them. I don't know that it'll solve all the problems in AI. I mean, right now we can't do all the things that humans can do. Look around you and you can see. I mean, we have office buildings full of people. And many of them aren't using their hand eye coordination. It isn't mechanical engineering which is a problem. You might ask, why are they there? What are they doing? Well, they're having meetings, they're filling out paperwork, they're doing studies, they're communicating with other humans, they're making plans. And those are things, why can't computers do all that? Why is there anybody in those buildings except the janitors and maybe a few top bosses? Well, and

computers would be cheap if we could do it, cheaper than those people. And so, why are they there? Well, because computers can't do it yet. And will lots and lots of data solve that problem? I don't think so. Might help, might be part of the solution, but the reason we don't have what you might call human level intelligence yet is that we just don't have the ideas needed in order to write the programs that would allow us to achieve human level AI. But, we have a lot of smart people and I think we're making some progress.

It will be a problem in the end, I think, for society what happens, what do we do with all the people that computers replace? And eventually, I mean, right now you need more and more skills in order to have jobs. But, there's this guy Robin Hansen, you know about Robin Hansen? Robin Hansen is an economist and he has got this interesting metaphor of sea level rising. Sea level is what computers can do. And land and the land that's inhabited, the jobs that require humans to do. And sea level's been rising. And on the shore a lot of people are displaced. Well, they've had to move to higher levels. But, to move to higher levels they have to have more training. Now, the fact that sea level's rising itself makes some higher levels. It's a funny thing. At sea level build some mountains so people can climb those mountains, but sea level will keep rising. And the question is will it rise above even those mountains? And so, what do we end up having people do?

Well, there's certain jobs that only people can do. You can't have a machine make sweaters made by hand. And there's kinds of things which involve social interaction which only people can do. Some of the social interaction maybe machines can do. But, if you really want a human you got to have a human. And so, I'm not saying that all jobs will be replaced, but we're already seeing a trend of many and I think that trend will continue and you read people who talk about the current slow recovery, the economic situation, and many people say "Well, we've laid off a lot of people because of the recession, but in the meantime we've found out we could do some of those jobs that those people did with machines and by the way, we're not going to hire those people back." And so, I think that's going to be a continuing difficulty.

**Q:** What's your advice for young people who are interested in a career in robotics or AI?

**Nils Nilsson:** Right now, I don't know, I think I'd try to get involved in the bridge between AI and neurophysiology. Let's take neuroscience in particular because I think there's still a lot of secrets about how the brain works that we don't understand that'll be helpful in engineering. The reason we don't understand them, I think, is we haven't invented the concepts needed to understand them. I have this analogy with computers. If you had a Martian coming down looking at computers, measuring all the currents flowing back and forth from the transistors, no amount of all that measuring, no amount of understanding how a transistor works is going to tell you how say an online banking system works or how an airline reservation system works. You have to have concepts that got invented in computer science to even understand them. You have

to have concepts like lists, programs, data structures, compilers and there are probably a whole set of analogous concepts in helping us understand the brain.

So, yes it's important to understand how a neuron works and a lot of neurophysiologists might complain that models of the neurons that the AI people are cooking up aren't accurate enough. Well, that might be, but no amount of detailed understanding of an individual neuron or even how neurons get interconnected I think will be sufficient to have a good explanation of how it is that we do what we do. We have to have some higher level things and there are people working on that. But, some of it, I think, will get developed by those very people who have one foot in artificial intelligence and one foot in neuroscience to say, "Ah, the analogy to these high level programs is such and such to the way the brain does this." And they will invent concepts that will then help us understand the brain better.

**Q:** Okay. Is there anything you'd like to add?

**Nils Nilsson:** I think you've got everything. Well, I think of things after you leave.

**Q:** Oh, well, on the wiki page is said that Charlie Rosen started a winery.

**Nils Nilsson:** Yes, he did. Ridge Wines, do you know about Ridge Wines?

**Q:** Yeah. I bought a bottle on my trip, but I didn't drink it yet.

**Nils Nilsson:** Uh-huh. Good wines, I've been up there helping him pick grapes.

**Q:** In Santa Cruz, yeah.

**Nils Nilsson:** Well, in Santa Cruz Mountains. It's actually in the foothills behind Cupertino.

**Q:** Okay.

**Nils Nilsson:** You can get to it, Ridge Winery, it probably has a website. It was bought, I think, by a Japanese company, but I think they kept the same winemaker and he and three other people at SRI started that winery. I think they've all passed away now. Well, everybody has, Charlie did too.

