Oral History of Raj Reddy, Part 2 of 2

Interviewed by:
Hansen Hsu

Recorded November 23, 2021
Mountain View, CA

CHM Reference number: X9458.2021

Hsu: So today is November 23rd, 2021 and I'm here for our second session with Raj Reddy. My name is Hansen Hsu. So I want to get back to you-- one thing that you brought up in previous interviews is you've talked about your definition of computer science, and how that differed with some of your contemporaries. Can you talk a little bit more about the difference between those definitions and how that may have driven the work that you've done in your career and the direction that you've gone?

Reddy: When you look at the emergence of computing, going back to Charles Babbage and the early machines, they were doing it because they needed computational power to do numerical calculations and so on. And at that point, there was no specific definition of computer science. So around 1965, when the computer science departments were being formed, there was a question. What is computer science? And the main thesis was, there are computers. There are lots of things happening around the computers, the phenomena. And computer science is the study of the phenomena of computing-- around computing. I was a graduate student. I didn't like that definition. I didn't dislike it, it's just that I didn't think it captured the big vision. And my definition was to equate computer science to all of engineering. In engineering, different fields of engineering enhance capabilities, physical capabilities of the human beings. I'm saying in the future, the role of computers and computing will be to enhance all the mental capabilities. Anything we do with our brains, not with the rest of the physical body, will in fact be enhanced through the use of computing. And so the whole spectrum is what does that mean? At that time, I didn't have any better definition. But looking back now, there are lots of things we do using our brain. At that time, we thought intelligence did not include walking, speaking and seeing. But now we know it's actually an integral part of that. And 80 percent of the brain is occupied by these activities; walking and talking and speaking, and so on. So the bottom line for me was that any activity that human beings are involved in, in the future will get enhanced. They'll be able to-- just like we had a car that made it possible for us to go ten times faster to twenty times faster, or a plane, 100 times faster, anything that we do mentally, using our mind, we will be able to enhance. Do the things quicker. Now, in retrospect, it's even further beyond. It's what you might call superhuman capabilities. There are things that we don't know how to do, and we are able to imagine that we will be able to do those things in the future.

Hsu: Thank you. Did the computer science department run into any conflicts with other departments or disciplines over what was within its purview?

Reddy: Yes. Basically, in almost every university, there was this big debate. And some universities, like MIT and Berkeley, where electrical engineering was extremely strong, they said, "Computer science is part of electrical engineering. Just one more aspect of electrical engineering. And therefore, we don't need a separate department. We will be it." And that has continued for 60 years, almost until 2018 or 2019, when MIT finally said, "We are going to do more than just calling it EECS. We're going to create a College of Computing with a billion-dollar endowment, which will actually interact with every discipline. Any discipline that's doing some intellectual endeavor will be enhanced through the use of computing. And therefore, we need people in all of them. And this was kind of obvious, even in 1985 or '89. I was at a National Academy meeting with David Patterson, and I said, "Computing will impact every field, including music and drawing and art and everything." So he came up with this idea, Computer Science and X, for every X. For all X. And I said, "That's it. That's exactly it." And so it turned out in retrospect, Stanford now has a CS and X programs for every X and something. Whoever wants to set up a joint program, it’s already streamlined and they know exactly what has to be done. So you-- any department can say, “We want to set up a CS and X.”

Hsu: And so you also carried that over to CMU as well?

Reddy: Absolutely, yeah. CMU and Stanford had no problem setting up a separate Computer Science Department. But those departments were in the College of Science at that time. And around 1990 at CMU, we said, “It is more than just a College of Science. It’s all of the university, every activity. So we essentially split off the Computer Science Department and created a college of Computer Science, a School of Computer Science. And that happened in 1990. We were the first to do it, and since then, many others have also done it. But the idea, it’s obvious in retrospect, but at the beginning it was not. It was not even clear if there should be a department [for] there’s such a thing called computer science.

Hsu: Right. Because originally it was either in mathematics or in EE.

Reddy: Yes, at Stanford was in mathematics. When I went in there to do my PhD, I had to do all the requirements of a math department. I said, I didn’t come here to do math. I came here to do AI and computer science. So finally, I kind of didn’t want to go through all the more detailed, formal mathematics. So I was actually going to switch to operations research at Stanford. And then my advisor, John McCarthy, said, “Don’t do that, because we are about to form a department.” This was in Fall of ’64. So I stayed on, and at the end of the year, we formed a department.

Hsu: Yeah. You mentioned operations research. That’s interesting, because Herb Simon at CMU was also very much in that area, right?

Reddy: Yeah, basically there were hardcore computer operations research types; Dantzig and Cooper and Henderson and linear programming, Simplex method and so on. But at that time-- but interestingly, Dantzig came to the Computer Science Department at Stanford around 1969. And it was very-- in ’68. That was very good. So it kind of-- they were all more new fields, right? Computer Science, Operations Research and so on. And we had to find a home for them somewhere. And it worked out.

Hsu: So you mentioned that CMU created the School of Computer Science in 1990. So essentially, when you became the dean in 1991, you basically were the first dean of the School of Computer Science at CMU?

Reddy: I was the second dean. Nico Habermann was the department head in Computer Science. And I was running robotics, a separate institute reporting straight to the provost. And when we decided to form a School of Computer Science, it was natural that Nico Habermann would be the dean. And then Nico came to me and said, “Do you mind if we merge Robotics into the School of Computing instead of leaving it as it is?” I said, “Absolutely. I believe in it.” So we merged that and the entity called Machine Translation Center-- Center for Machine Translation, Language Technology Institute. Which in retrospect became very big. All of Google and everything else is human language technologies. Understanding and translating and summarizing human language turns out to be central, and it has created trillion-dollar businesses, in retrospect.

Hsu: So that was also a separate…

Reddy: Entity. We merged all the three of them into the School of Computer Science. And after that, in 1990, when we added another three or four more departments. We added Human-Computer Interaction as a separate department, I think in ’92 or ’93. We changed the name from Center for Machine Translation to Language Technology Institute about the same time. And then, we created a Center for Data Mining and Discovery, which became Machine Learning Department in 2005 or so. And then most recently, around 2007 or ’08, we added a Department of Computational Biology. That’s the first one where we have an old, existing discipline kind of merging with computation to create-- so that basically will become the norm for almost every discipline. Computational mathematics, computational physics, computational chemistry, the usual departments, but also computational music and computational drama. Everything will add a computation in front of it or in the back of it. And then suddenly there’s a new area of study. How you use computational methods, technologies, to discover things that were not discoverable before. That’s what all deep learning is telling us. There are patterns of knowledge in the data that you can find from big data that cannot be discovered by the human mind. And for the first time, that has proved to be very important.

Hsu: Yeah. So what sorts of challenges did you face when you started as the dean? Or what sorts of changes did you oversee? Well, you just mentioned some of the changes, but what are some of the other ones?

Reddy: I was very fortunate. Basically, I walked into a very supportive environment. I was the dean. The provost and the president were completely supportive. And so anything I wanted to do, they said, “Okay.” That is unique about Carnegie Mellon. MIT and Stanford have enough structures and bureaucracies and rules and regulations. You cannot just create a department. So I singlehandedly said, “We’re going to have a department of Human-Computer Interaction.” I didn’t ask anybody. I just declared it. And that was it. You couldn’t do that in almost any other university.

Hsu: So after your tenure ended in ’99, how did you leave the colleges differently from when you left it, other than starting all these other departments?

Reddy: Yeah, basically, two or three important things happened. Number one, we have a college now that’s six departments. The seventh came after I left. And secondly, our size. When I started in this Department of Computer Science in ’69, we were ten faculty members and 30 graduate students. That was it. Very small. And by the time I became the dean, we were probably 30 faculty members and 100 students or something. But we had a Robotics [Institute] that was kind of doubling every few years, eight years or so-- seven or eight years. and so by the time I started as the dean, all these programs were growing very rapidly. And in particular, Robotics became much, much larger. So much so at this point, now 40 years later, 30 years after the forming of the school, we are now 300-plus faculty members, and 2,000 students. And half of them are graduate programs, master’s and PhDs. So it’s kind of changed significantly. I don’t remember-- it’s not so much when I left. It’s the continuous acceleration of the growth of the department. So around that time, we started a minor in computer-- we started-- a undergraduate program in Computer Science was started around that time.

Hsu: What year was that?

Reddy: That was around ’91.

Hsu: ’91, okay.

Reddy: ’92. And we..

Hsu: So that late?

Reddy: Yes.

Hsu: Wow.

Reddy: Because we were only a PhD program. And they wanted us to change it, but we said, “In order to do that, we need a lot more resources.” And they said, “We don't have any resources.” So ultimately there was so much pressure from the students, because they said, “We don't want to get a BS in Mathematics or Electrical Engineering. We want a Computer Science degree.” And so ultimately, now, we have two or three different undergraduate degrees in the School of Computing. We have an undergraduate degree in Artificial Intelligence. We have an undergraduate degree in Human-Computer Interaction, in addition to an undergraduate degree in Computer Science.

Hsu: Wow.

Reddy: And all of them are kind of growing. And it turned out there’s so much demand, many of the students we rejected are admitted into MIT and Harvard because they have a much larger pool. So in the last two, three years, CMU also relaxed the size restrictions in the School of Computer Science. And now, we are almost 50 percent larger than two years ago.

Hsu: Wow.

Reddy: But mainly because there is so much demand, we were rejecting people, very high-caliber people, on some trivial grounds. And that didn’t make sense.

Hsu: Wow. Let me skip over here. Let’s go back in time a little bit and let’s talk about the Stanford Arm and the demo that you did in 1968.

Reddy: Right. What about it?

Hsu: So were you-- how did you get into that project? And was the Arm already going before you joined the project?

Reddy: Yeah. So basically I was part of the AI group or AI Lab headed by John McCarthy, and Ed Feigenbaum was the deputy director of the lab. And so I was a student until ’66, was the first PhD in AI and Computer Science in 1966, and Stanford hired me as an assistant professor. So this was ’66. During that period, we had three different research groups, among other things. One of them was speech, a computer that you can speak to and took back commands. The second one was vision, to see and understand and analyze an image and detect the location. And the third one was a manipulator that could do things. So all these things were already at work in 1966. And by 1998 [1968], all of them were doing by themselves.

Hsu: By ’68?

Reddy: Yeah. At that time, Les Earnest, who was the kind of executive director of the AI Labs there, said, “Why don't we put it all together into a demo?” And we thought we could do it in a week, as usual. It took us six months or a year, because combining all the different programs written by different people in different ways to kind of do a single demo was kind of interesting and difficult. So if you look at that video, “Hear! Here!” video, you’ll find the names of various people in different projects, working on different-- but they’re all there. And some of them are actually in the video. And that’s how it happened.

Hsu: Oh, I see. So three different projects have merged to...

Reddy: Three different projects combined together to illustrate progress in AI.

Hsu: And what went into the making of that “Hear! Here!” video?

Reddy: What went into the making of what?

Hsu: The making of the video. Can you talk a little bit more about the making of the video?

Reddy: Yeah. Basically, there was a culture in the AI Lab to create short videos to demonstrate-- the video technology was not there. There were no VCR’s. Nothing was there. So to demonstrate the thing, we would make a 16-millimeter film, right? And so I believe there were one or two small demonstrations using 16-millimenter film. And there was a group at Stanford Communication Department, the film-making department, which was interested in getting some of the people and doing something. So we had a natural confluence, and they came over, and then we worked together. The idea is to make a technical documentary, which is easy for ordinary people to understand without diluting the technology of what it’s trying to do. So it was kind of an interesting challenge. It still is, that is the biggest challenge, and no matter what area you pick up in AI or computer science, making it understandable by non-experts, people in-- ordinary citizens, becomes very important.

Hsu: Yeah, that's definitely still a challenge. Let's move on a little bit. We talked a little bit about Hearsay last time, but I'd like to go back to the beginning of the project. Tell us about how Hearsay began.

Reddy: Yeah. So basically, at that time, everybody believed there is either knowledge on how to do something, or no knowledge. If there's no knowledge, then you have to kind of do trial-and-error, like a puzzle or chess. You can play chess without any knowledge just by trying out all the different moves. And so the idea of knowledge-based systems was in the air. And the first major demonstration of that happened with Ed Feigenbaum's Dendral project. And what he said was, "Here is a major research area of interpretation of x-ray crystallography from looking at-- to understand-- to derive a molecular structure of the molecule that is being crystallized. And there were PhDs in biology who spent a lifetime studying those images of x-ray crystallography, and could infer that molecular structure. And Feigenbaum and Carl Djerassi and two or three others basically said, "We should be able to understand exactly what they're doing and capture that knowledge into rules, and then let the rules determine what structure might be there based on the data they're looking at. So this whole area of knowledge-based systems, where you're extracting the knowledge and [a] knowledge engineer with someone who talked to the experts and extracted the knowledge and put it into a rules-based system, which rules when applied to the same data would result in same or similar result. So that was the broad, accepted theme for almost all of '70s. And so in AI was knowledge-based systems. If you had knowledge, you don't have to do trial and error. If you don't have knowledge, you have to search, like to solve a puzzle with all the different-- so the idea of Hearsay for speech was the different kinds of knowledge in understanding speech. For example, there's knowledge about acoustics and phonetics and linguistics and syntax and semantics of the task. And more broadly, common sounds and pragmatics and so on. All of that go into understanding a spoken sentence. So the question was, can we capture each of these kinds of knowledge? And then based on these component knowledges, infer what was being said. The challenge was the inferring of what was being said required getting all these different knowledge sources to talk to each other, or somehow collaboratively solve a problem. And the analogy we used was supposing we took ten scientists from ten different countries, each of them speaking English or Japanese or Chinese or German or something, but they have to solve the same problem. And so we came up with this model called the Blackboard model. The Blackboard model was whatever you infer, you write down on a blackboard what you inferred. And the others can see what you are hypothesizing. They may not understand your language. They may not understand how you came up with that hypothesis. All that they're seeing is the hypothesis that I saw these sounds or I saw this word or something. And say, "If you saw these sounds, then I will postulate it must be this word, right? It's one of these three words. Maybe out of all those words, the syntax and semantics would rule out a lot of the options. And the way that whole area of how different knowledge sources impact the ultimate recognition. So what we had to do was sometimes in what we call ablation experiments, we took out the whole knowledge and said, "What happens now? So we said, "If you don't have syntactic knowledge or semantic knowledge, then you're not going to-- you'll come up with some-- you have to work a lot harder to come up with an answer. And when you do, sometimes it may be wrong. And that was the whole idea of how do you solve problems using multiple knowledge sources rather than a single knowledge, like in the case of x-ray crystallography. And that turned out to be an important model. And the Blackboard model was then used-- and the psychologists love it and they use it all over. And Feigenbaum and Penny Nii and others used it for some of the other things. And so it was an important model at that time. And the main thing is-- here is the basic discrepancy. At that time, at

Carnegie Mellon in ’85, we had a young professor called Geoffrey Hinton, who then went on to Toronto and did the deep learning stuff. And his thesis was, “Look, there may be knowledge. I don't know what it is. I don't even want to worry about it. I want to build a system which will somehow learn by itself. I don't need to know it. I just need for the system to figure out by itself using...” When a human child is born and over a period of time, they acquire knowledge, they can’t articulate how they recognize the speech. They can’t articulate what makes me look at an image and say, “That’s a door and that’s a person,” and so on, because they do it by osmosis. So the question was how do you build systems that learn from experience. Don't worry about knowledge. Don't worry about the knowledge-based systems. So that was a dichotomy. Basically, one group said when you see a phenomenon, you try to create a model or a theory of what it is and how it works. Then that leads to a set of theories in physics or chemistry or biology, which then you apply to further elucidate the situation. The other one says it’s a black box. I don’t care, as long as it’s able to learn by itself and discover all the details, I will let it discover. So it’s turning out [that] our human inability to infer all the rules and all the knowledge, given all the data, is profound. And therefore, the performance of systems where you have to create a theory or a model--a hypothesis and test is actually a theory. If somebody says, “I postulate this is the word that is there,” and it may be right or wrong and the test and verification process will rule it out, perhaps. And that was the main result of that time. And so I would say at this point, we are still up in the air. We know just by using deep learning and machine learning we can learn lots of things and actually do things. However, we also don't yet know how to use that methodology and that technology to learn things that human beings seem to learn. So for example, I can show you some example and say, “This is this,” only once. I don't give you a million examples of that. The single example, what is now being called one-shot learning and a few-shot learning, you can actually learn the stuff. And people are beginning to experiment with those concepts. There are other things that we do. We learn from teachers. A teacher kind of gets up and says, “This is true. It is how you do the addition or multiplication.” And then you learn from that. And we learn by mistakes. When you make a mistake. And so all of those kinds of learning are not yet practicable in deep learning structure. But I’m hoping we will come up with new methods and new methodologies that’ll encompass all of that, within the overall framework of learning from big data, right? We don't yet know how. You cannot form theories, or you cannot form knowledge without data. And then the main thing that’s currently missing in the deep learning methodology is when you learn some knowledge, that must be transferrable to somebody else. In psychology it’s called chunking, when you keep learning things-- A, B, C, D, E, F, G-- and it becomes one chunk. Now, if I tell you to say the same thing in reverse order-- G, F-- we have never practiced it, so it’s very difficult to say backwards. So we have to think, and work out the thing. And so it takes much longer. And this is what Daniel Kahneman, who got the Nobel Prize-- so thinking fast and thinking slow is about mainly when you take some knowledge and form a chunk, then you can retrieve the whole chunk at once. Don’t have to-- and we don't yet know how to do that in deep learning. We do form chunks, and we do actually use them, but what we haven’t yet figured out is how to kind of make that into a piece of knowledge that other people can use. If everybody has to go through the same experiences and learn the same thing, that may be more difficult.

Hsu: Getting back to Hearsay, so Hearsay was the first application of knowledge-based AI in speech understanding?

Reddy: Yeah. Basically, even as we were doing it, there were people like Jim Baker, who did the Dragon system, were already coming into our group saying, “We need to begin to use statistics.” We agreed on that. Not necessarily deep learning type statistics, but using statistics like base rules and condititional probabilities and so on. So there, the statistics were used narrowly. Narrowly in the sense of, only the acoustic data was statistically modeled to be able to infer what is being said. That’s what we call signal-to-symbol transformation. But from there, going to all the other steps-- word formation and phrases, recognition-- we didn’t do anything more. But the whole thing was already-- so we were using statistics and distributions and so on in speech as early as ’73, ’74. Jim Baker came there to CMU in ’73. So my suspicion is-- and Kai-Fu Lee’s thesis demonstrated the power of it-- but what he said was until that time, people built systems that are what we call speaker-specific recognition systems. You had to train it with your voice, and then it became good. And he said, “You don't need to do that. Give me enough data, I will train it for all the voices, and I will statistically model what is being seen, and then I’ll build a speaker-independent recognition system.” This is what he did. That was in ’87, right? It took from then, ten years. And so all these systems simply said yes, at a broad level, there are all the different kinds of knowledge. But at the lowest level, modeling of sounds, of what is being said, has to be done statistically. And that’s what we did. And it worked beautifully, and then-- but we never…

Hsu: The statistics?

Reddy: Yeah. We never said we’re going to learn all of that from raw data, semantics and syntax and all the other stuff. And the current effort in both speech and language understanding and image understanding is to kind of assume you don't know any of it. Learn all of the levels completely from scratch.

Hsu: Okay. So even in Sphinx, the other levels were still-- there was still a model, there was still...

Reddy: Exactly.

Hsu: Okay. That somebody had to hand-tune or construct?

Reddy: Exactly.

Hsu: Okay. Interesting. So yeah, you’ve already touched on this. I kind of wanted to hear you explain exactly the ways that the three systems in the '70s-- Hearsay, Dragon and Harpy-- differed from each other.

Reddy: Hearsay was using the Blackboard model. What Jim Baker did was he said, “I need to be able to search this space. So I’m going to map it into a single graph.” So what he did was he took the words, and then mapped each word into sounds in the lexicon, and then created a bigger graph. That’s what, in the Harpy video, it shows. And the word graph can be converted to a sound graph by simply replacing each word by that component subgraph of the sounds. And then what the Harpy system did is what-- in Jim Baker’s system, you had to kind of do exhaustive…

Hsu: Harpy or Dragon?

Reddy: I'm sorry. Hearsay and Dragon and Harpy. Now, Hearsay was a Blackboard model. Dragon was a graph model. And it had to search the graph and search every path and calculate the probability it could be one of those. As long as you can represent what you can say in a huge graph, a finite state graph, then you can calculate the probabilities of every path and find the path with the highest probability and say, "That's what you have said." And at that point, it was clear that's fine. It's just that doing an exhaustive search of every path was taking an hour for every sentence. In those days, the computers were too slow. And that was not acceptable. We needed a real-time system. Something that would work with it. So what the Harpy system did is, I'm not going to do an exhaustive search. I'm not going to give you the best path. I'm going to find the good-enough solution, which may be correct, which may be occasionally wrong. I'm not finding the optimal solution. This happens in economics and statistics, everywhere. People want optimal results. And then there are others that say, "Human beings don't optimize. They satisfice. They just find a good-enough solution." Because when you're traveling on a path, going from San Francisco to San Diego or San Jose, sometimes you may take a different path for no good reason at all. You're somewhere-- and so exactly which path you take may appear to be-- even if it's not optimal, it's good enough. And human beings do this all the time.

Hsu: It's kind of like bounded rationality, right?

Reddy: Yes. And basically what bounded rationality of Herb Simon says is that human beings, given all this knowledge, all this data, are incapable of analyzing everything to come up with an optimal solution. So what they do is bounded rationality or bounded-- essentially find a good-enough path, a good-enough solution. That's what Harpy system did. We introduced the concept called Beam Search, where at every stage we would say I'm not going to look at a thousand different paths. I'd look at top ten percent. I'll throw away the rest. I'll assume if the probability is pretty bad, it's probably not going to be a good thing. There may occasionally be a path where it actually is the right one. You find this now when you're dictating into Siri or messages. Sometimes it'll recognize something, and then at the end it'll come back and change some words. And you might say, "What in the hell happened?" Because it actually then recomputed all the probabilities using the latest language models, GPT-3, and says, "Oh, that must be-- this is more likely to be that." So that's, I think, what happened with it.

Hsu: Right. Okay. And so you mentioned Hearsay used the Blackboard model. That was an advance in the second version of Hearsay, correct?

Reddy: Yes. No, even in the first version we used it. Except in this first version, it was more of a simplified Blackboard model. And the second version, Hearsay II, we had people like Lee Erman and Victor Lesser and Frederick Hayes-Roth, who brought a lot more brainpower to the problem of how does a set of independent agents, how do they interact? Because people are putting in something. How do you focus of attention? What do you look at first and what do you look at next and how do you-- there are a whole bunch of issues in a distributed system of that kind. In fact, if you look at Victor Lesser's subsequent career, it is all about distributed agents. How can a group of agents that are kind of doing different things interact with each other and find solutions? It's a very interesting area.

Hsu: So what would-- is that what you'd say was the most significant advance from Hearsay I to II?

Reddy: II, yeah.

Hsu: Okay, is the-- just more sophisticated, distributed agent management...

Reddy: Distributed agents, yeah.

Hsu: Okay. So that part wasn't there in the first version?

Reddy: No. Or it was there in a very simplified manner. You essentially say, "I don't know how all these things work, but we will pick this particular alternative and see where we go." And even things like, for example, in Hearsay II, the video, we use-- there was a lot of debate about semantic models. How do you-- what is semantics? Where does it come from? In any given situation, how do you use semantics? So it turned out if you used the task of chess, you know the chess board and all the possible alternatives. And you can't just use the words "pawn to king four." You can't say, "Four, pawn, king," because there's a structure that has to be said. And even if you said-- you can't say, "Pawn to king five," or some other thing, because that's not legal semantically. It's not a legal move. And so what was happening was syntax was constraining what words you can use, and semantics was further constraining what sentence structures by eliminating all the sentences you could not say. Because these were only-- there are only 30 legal moves. You can only say one of those. You may say it a slightly different way, "pawn to king four," or "to king four pawn" or something, some reverse order. That may be legal, but lots of other things are not. And so that's what was done.

Hsu: That works for a very bounded area like chess, but for regular speech, that's too much...

Reddy: The assumption is it works at everything. That is, human beings, when we come in-- when you say suddenly, out of the blue, "Do you want some coffee?" And there's a whole amount of context. It's in the morning and people come in and they might want coffee. It is the social behavior. There's a whole set of things that have nothing to do with, "Do you want coffee?" And to determine-- you could want any number of things, a million things. To determine you're talking about coffee means you need to understand the semantics of the situation. It's just that we were lucky enough to find a task like chess to demonstrate the power of semantics. And at that time, we were all talking about syntax, semantics and so on. We didn't know what it means. And what we did know by the time we were done is each of these knowledge sources is constraining the alternatives and saying, "Of all these things that you've put up, I'm not going to consider any of them, because these are not likely. They cannot be used." And therefore they are automatically rejected. And even if they may be used, if they are of very low probability, then it may still be rejected. And so there's a whole set of interesting questions of that kind, where you might say, "Do you want to have breakfast?" Okay, eggs, bacon, all those words. You might say something like, "I want to eat kiwi fruit" or something. That's something very unlikely and rare, but you might say it. And how the question is, how do you deal with improbable words when they are actually being said? And most current systems simply reject it. And they won't even give you the thing, and that's when you get wrong answers. And this usually happens in Siri--like dictation with names. You say an Indian name or a Chinese name or something, it doesn't know how to deal with it. And so it'll make up an English name and you have to physically go and manually correct it. Yesterday I sent a message. I said, "Hansen, I'll be there tomorrow morning." It misspelled your name. Instead of E-N it was spelled A-N. If it was smart enough it should have seen the message came from Hansen with an E and would have put it in. It wasn't. And so the net result was there was a mistake. I said, "He will understand. I don't need to correct it." I just sent it. So it turns out we tolerate a lot of the error in text messages because people understand it's not that you didn't know the name or the spelling, it's just that it was not worth the extra time to correct it. Hansen or Hansan, it doesn't matter how you spell it. You knew what I was saying.

Hsu: Going back to these early systems-- Hearsay, Harpy and Dragon-- what sort of hardware did they run on?

Reddy: All of them were using PDP-10s, because that was the mainframe computer we had at Carnegie Mellon. And at some point in 1975, we were coming close to a major demo with DARPA on the thing. We needed a lot of computing power. So at that time, we ordered a second PDP-10. Allen Newell, bless his heart, said, "We're not going to put it in public distribution for everybody else to use the second PDP-10. It'll be only for the speech group for one year. After that..." So we had enough computing that-- in the history there, there's a story in Bell Labs when Bishnu Atal needed a supercomputer to debug his cell phone protocol. And he used up the budget of the whole acoustics department in one day. And so the question was, what do you do? And to their credit, the director at that time, I think Arno Penzias, said, "Well, we're just going to buy a computer. Another supercomputer. We are going to call it lab equipment and put it in his lab for a year. After that, we'll figure out what to do." So people understand the requirements, needs of researchers. When you're hot on a trail, you cannot be restricted by rules and budgets and other things. It's an amazing thing. That is true in the United States. Not true in most other countries, where they are all rigid. They have rules, they have to follow the rules. Germany, for example, is impossible.

Hsu: And what sorts of programming languages were used for these systems?

Reddy: At that time, we had-- we were using LISP. LISP was too slow, so we had to write most of the things in programming languages of the day that was available for PDP-10. I forget now. There was a language called BLISS. I don't think we used BLISS, but we used some of the other languages. And the one we used was some variant of ALGOL that was actually done at Stanford by Dan Swinehart, and we imported it and used that.

Hsu: SAIL?

Reddy: Could be. I forget the name. Yeah. I think it may be SAIL.

Hsu: Okay. That's interesting. Let me move on a little bit to-- because I think Alan Kay told us that Bruce Lowerre's involvement in speech was purely accidental. Is that correct?

Reddy: No, it was not so much accidental. He was a Ph.D. student, and I said to him, “Your thesis is to figure out how to make Dragon system 100 times faster.” And he was kind of trying to look into things. And then we both looked at where it was actually trying all these alternatives. And at that point, that was the bottleneck. And I said, “Bruce, I think you should just ignore 90 percent and follow the rest.” And then he implemented that and defined it. So that’s what usually happens with Ph.D. students. Namely, they are the ones that take your little bit of an idea, which may work, may not work. They take it and run with it, <inaudible 00:57:16>. And he was the one that did it.

Hsu: Right. Can you talk about Alex Waibel and Janus?

Reddy: Yeah. Alex Waibel was one of our Ph.D. students who came from MIT. And unlike others, he needed a job. And I said okay. We gave him a job. And so he was a part-time student and full-time employee. And he was working on a number of things. And at some point, he was no longer part of the speech group I was running. He had other funding and he was doing other things. He was also a professor at Karlsruhe, so he had a dual appointment. An appointment at Carnegie Mellon and also an appointment at the University of Karlsruhe in Germany, where he is from. And he had a lot of students there working on the same thing. And at that time, we were trying to build systems that can take in-- and it came in because of the problems they were experiencing among the troops. They would land in-- not Belarus, in Yugoslavia, one of the different-- half a dozen countries there, each speaking a different language. And the soldiers didn’t know how to speak the language, and they needed even simple translation. So the question was can you build a system where you can speak in English and the system would translate the simple phrases into the local language? And so one of the systems that was built was by Alex Waibel to do speech-to-speech translation on a very limited scale. Even now, there are systems that do-- I speak a whole lecture or something, it will translate it into-- English to Chinese, English to Japanese and so on, and other languages. But it is not that great. It makes some mistakes. But more importantly, the errors it makes are not directly attributable to the inherent language. It’s mainly because of the vagaries of the system implementation. So we can actually demonstrate it. There’s an app these days called-- I forget now. I’ll look it up on the phone, where you can actually speak in any language in a meeting and it’ll translate it to the individual languages of all the other people automatically, but not that good. It makes lots of errors, but it’s good enough so if you already know the subject of the meeting and what other people are saying, you know the language a little bit, you understood most of it. And then you still-- if you had access to it, it’s easier. And that’s what has been an interesting...

Hsu: So how did Janus work?

Reddy: That, I don't know. Broadly I know, mainly. It’s just a question of recognizing the speech, translating this phrase and then synthesizing. That means you have to build models of text-to-speech synthesis, speech-to-text recognition, and translation. Each one of them, if you limit the phrases that you can say, you can do a reasonable bit of translation. But how specifically they implemented it and languages, I don't know that detail.

Hsu: Can you talk about working with Takeo Kanade?

Reddy: Yes. Takeo was one of the early Ph.D.s in computer vision in Japan. He was a Ph.D. from the University of Kyoto. And his thesis was on face recognition. I had a student at Stanford at that time, Mike Kelly, who was also working on face recognition. And the implementation of Takeo was much better. The results were much better. And so I was at a workshop in Kyoto in 1973 when I saw his work. I said, “This is great.” So a year or two years later, I invited him to come to Carnegie Mellon. And then he came and spent a year or two years and then went back. So again in-- this time around I said, “Why don't you just come back here permanently?” And so I think it was 1979 when he came permanently, because he was doing much better work in computer vision than I was capable of at that time, and I was not so much dabbling in it. It was getting more and more physics-oriented. You had to kind of understand the physics of imaging and reflections and all those things.

Hsu: So the entire time, you also had done a little bit of computer vision?

Reddy: Oh yeah. I’d been doing computer vision. Only after Takeo came to CMU fulltime, I said, “Takeo, you take over this area, because there’s no reason for me to duplicate what you can do with it.” So I stopped at that time.

Hsu: Right. And so he came after Robotics Institute had already started?

Reddy: About the time.

Hsu: Around the same time.

Reddy: The Robotics Institute was started in ’79. He came a few months before or after, initially on a temporary basis. Then I convinced him to stay on permanently. And then we gave him tenure at some point. So he was very central to all the computer vision works, because we were doing a lot of autonomous vehicle work also at that time. All of that needed computer vision. And so he took that.

Hsu: Right. So did you have any collaborations?

Reddy: We did. We had a number of early papers that we published together. At some point, I said, “You’re on your own. Do whatever you want to do.” And that worked out. He was really great.

Hsu: Can you talk about Manuela Veloso?

Reddy: Yeah. Manuela Veloso was a student of Jaime Carbonell. Manuela had already done advanced work in Portugal in computing and related areas. But she was an electrical engineering student. So she came in, and her husband was Jose Moura. Both of them came, and he came as a professor into electrical engineering department and she came into computer science. She did her Ph.D. with Jaime Carbonell on machine learning. And she was very good at that time already. But she was also a more mature scientist who’s maybe five years older than other graduate students. And so she was-- when she finished her Ph.D., Jaime recommended that we hire her as a professor, which we did. And then she graduated to full professor. And I think at some point, maybe 2010 or something, we elevated her to the rank of university professor. And then she became the head of the Department of Machine Learning at some point. Now, she's at JP Morgan as their chief AI scientist of JP Morgan. And doing very well. For a change, basically, when human AI researchers do work, you can't put a monetary value on it, how important it is. So when she was working on robot teams, playing Robo soccer, yeah, they said, "It's cute. It's interesting, maybe good AI research, but what do we care?" or something. And when she went to JP Morgan and applied the same ideas to money, she was taking and saving them billions of dollars. In particular, a simple thing like fraud detection, credit card fraud detection, turns out to be billions of dollars of business losses annually. And it's all assumed that they take that loss as given. And so even if you can eliminate ten percent of the losses, you would save significantly. So there are lots of areas in banking systems where enhancing the capabilities of any activity, making it better or more accurate, turns out to be worth lots of money. And that also is-- banking, like everything else, is a big business. The number of transactions per day they have to handle are in the tens of hundreds of millions. And they have to check every transaction, make sure it's a legal transaction before they can do it.

Hsu: Right. So is she still part of CMU also? Or is she fully at JP Morgan?

Reddy: She's on leave. I was saying she will probably not come back. At JP Morgan, she's head of a group of 5,000 people or something that work at different places. So it's difficult to decide. And also they probably desperately want to keep her, because she's one of the rare talents. And so it's unlikely CMU can match the salaries of JP Morgan.

Hsu: Yeah. So let's see, more about-- for the major areas of research at Robotics Institute. Obviously autonomous vehicles is a major one, computer vision pretty big. What are the other major focuses of research?

Reddy: Language technology and all aspects of language.

Hsu: So also part of Robotics Institute.

Reddy: Not part of robotics, but it was School of Computer Science. It was called Language Technology Institute. It was a separate department.

Hsu: Right. But Robotics Institute itself, what are the major areas of research?

Reddy: In Robotics Institute, we started with two major areas. One is autonomous manufacturing-- where the factories would keep running with no people, autonomic and so on-- and autonomous vehicles. Those were the two broad areas we started with in 1979. Later on, we added medical robotics. Hip replacement surgery and all kinds of things related to vision and whatnot. Then we also added a number of other components, design of-- for example, Howie Choset does snake robots. And design of different kinds of robotic manipulators on autonomous vehicles. We started earlier with three kinds of autonomous vehicles. There were land vehicles and there are autonomous sea vehicles and autonomous air vehicles. Another student working with Takeo Kanade built an autonomous helicopter that was used in the Three Mile Island-- not Three Mile Island, in the 9/11 Johnstown disaster. There was kind of-- nobody knew what to do because the whole area was blocked off. And so you had to kind of look at it. So the autonomous helicopter was going to be able to image and send back the data for the thing. And that was done by a student of Takeo Kanade. And the autonomous sea vehicles, we started some work because Office of Naval Research was funding us. But we were not near an ocean front, like Wood's Hole or something. So it never took off. We could have done it, but that means we would have had to set up a separate lab in some other location. Sometimes it's easy to do and sometimes not. So those were the early, early things. And then medical robotics came later. And then we started also an activity on applied robotics. Robotics for harvesting and robotics for going into hazardous areas like volcanoes, and robotics for-- it's called field robotics. Red Whittaker was in charge of that lab. And so they went to Atacama Desert, and then they also went to some volcano in Antarctica, where they tried to go and-- so there's a whole set of work of that kind, mainly coming from space research. Basically they wanted to simulate conditions in space to test out their ability of the robots to survive in that environment. So there are a whole bunch of things. And then the first time we used deep learning in some kind of-- is when one of Kanade—Chuck Thorpe's students actually used deep learning to learn to navigate. Collect the data, get the features from the data, and then based on that, navigate. So learning which way to go automatically at speeds of 60 miles an hour turned out to be important. The name of the student is on the tip of my tongue, but I can look it up later. And so that turned out to be the first major demonstration that you can build autonomous vehicles without having to physically program every step of the program. It would learn from experience. And that's what almost every system does now. The only thing that they don't have is there are things that-- not all experiences are there and they're not able to occasionally make a mistake. The second thing they're not able to do is deal with ethical questions, saying, "If you go this way, you kill one person. If you go that way, you kill two people. Except this person had nothing to do with it. He was just running around." Normally, you would not kill him, because you would just go plow into whoever is in the middle of the road. And by this strange decision you are killing somebody who is completely innocent. And there is a movie called Eye in the Sky which had to make the decision...

Hsu: The drone-- yeah. Drone warfare, remote-- yeah. <overlapping conversation>

Reddy: And so basically they had to make this decision saying whether to kill this little girl selling roti or something, some bread, or permit some major terrorist act had to happen. If you don't kill the terrorist at that point in time, which would also as a side effect kill the girl, instead if you let him go, he would go and blow up 90 people and was getting ready to do it. I think it's called Eye in the Sky. Something like that. And Helen Mirren is in it.

Hsu: Yeah. Alan Rickman, too. I remember.

Reddy: Yeah. So that's the kind of thing where you have all those issues are even now undecidable. It's up to-- somebody has to make a rule. And people will say, "What gives him the right to make up that kind of a rule?" Maybe the other one is right in this case. So it turns out—again, it's a question of context and semantics. I think there's an example in that book, 7 Habits of Highly Effective People, where there's kids in the subway running all over the place and not being controlled by the father. And the other people got all upset, saying, "Why is the father not controlling these kids?" And then he says, "We just came from seeing my wife die. It just happened. They're just not in a position..." So given that context, people are willing to forgive them.

Hsu: Can you talk about Chuck Thorpe?

Reddy: Chuck?

Hsu: Thorpe.

Reddy: Chuck Thorpe, yeah. Chuck was one of our early Ph.D.s, I think maybe in robotics. He may have gotten the degree in computer science.

Hsu: He was your student?

Reddy: He was my student, and then I handed him off to Takeo. So he's kind of our joint student. And then we hired him on the faculty and he went on to become the third director of the Robotics Institute. So Chuck was one of Takeo Kanade-- one of our students working in the autonomous vehicle area. And there are a number of videos of his-- you might have gotten some of them-- that has him and his students with Alan Alda or somebody, and they're demonstrating the whole thing. And after that, he became the head of robotics. And then when we opened the CMU campus in Doha, he was asked on my recommendation to go and be the first dean of the-- because he comes from a background where he grew up in Africa somewhere. And so he understands the issues that you kind of come up with.

<01:19:22 break in recording>

Hsu: We were discussing Chuck Thorpe.

Reddy: Chuck Thorpe then went on be the dean of the CMU Qatar campus, which we set up, I think, in 2003. And he was there for, I think, almost ten years. And then he came back. Then he was hired as the provost at Clarkson or some university up north. And I'm in touch with him maybe once every couple of years or something. Nothing new to talk about in the research domains.

Hsu: Right, because he's an administrator now. You mentioned Red Whittaker before. Can you talk a little bit about his work?

Reddy: Yeah. Red is very unique in the Robotics Institute. He's the one that has built a number of systems, robotics systems, that have gone on to have impact in a number of domains. And so basically Red tends to be nonconformist namely, and as a result, he was an assistant professor in the civil engineering department and they didn't give him tenure because he didn't have enough publications. So I said, "It doesn't matter." I knew he was very good. So I hired him into Robotics Institute in '83 or '84 as a research scientist, so I could just do it without having to get anybody's permission. And that was one of the best decisions I made at that time. And we had the work he was doing. And the other person that was interested in mobile systems is Hans Moravec. Hans was-- and I knew Hans from Stanford. He was a

Ph.D. student here and came to Pittsburg in ’82 or something. And I hired him to continue the work on mobile systems, mobile robotics. And he did a very good job. He continues to do a good job, but the number of working systems that Red has been able to build, and the cost, was significantly lower than almost every other researcher that I know of. And that makes him really unique. He understands exactly what has to be done and how to do it and gets it done.

Hsu: At low cost.

Reddy: Very low cost. The example I give is, when we got them both started, I gave Hans Moravec 200K and I gave Red 30K, because I wasn’t sure what the hell they were going to do. And for 30K, Red delivered more than the 200K. It’s not that what Hans was doing was not good, it’s just that he was very thorough in getting various things done, whereas Red knew how to engineer systems to make them work. A very unique capability.

Hsu: Hans has made a pretty big impact, too.

Reddy: Oh yes. Oh yes. For example, the books he has written, the Robot and subsequent books, have had a major inspiration on [the] Singularity. For example, he projected-- that was the first book that projected that we will reach Singularity by 2040. That we will have computers that are faster, more powerful than the human brain. And Ray Kurzweil, a person who kind of took up on that idea and then expanded on it substantially. And in fact, yes, I think later Ray had written a book called *Singularity*. And Ray turned out to be a major, major figure in AI and AI research on building systems. He also had a kind of different path and career. Whereas people like me are kind of traditionally-- we go into academia and we stay there, whereas Ray has built several companies and sold them. I think now he is working for Google or some fellow or something. I don't know exactly where he is and what he’s doing, but I’m sure...

Hsu: Well, one of his companies—Kurzweil’s companies was speech, right?

Reddy: Yes. One of the-- and he sold it to Lernout & Hauspie, who then bought Dragon Systems for significant money. Unfortunately they and also Bakers company went through this Enron problem where the books were cooked. And as a result, Bakers lost most of the money that they made.

Hsu: Oh, I didn’t know that, that there was a...

Reddy: Yeah.

Hsu: Wow, okay.

Reddy: Yeah, Dragon Systems was sold to Lernout & Hauspie, and instead of taking cash, which they were offered, they took it all in stock. They could have taken 50-50 or something. But in retrospect, it turned out that they would have been better off. And there’s nothing you can do when somebody cooks the books. There’s no amount of scrutiny that will tell you, especially if you are mainly a technical person like Jim Baker and Janet Baker and you're not used to all the shenanigans that people can-- accountants can do. And so ultimately it blew up in everybody's face. Part of the issue was Goldman Sachs should have given them the right advice. They didn't. And then when they went to court, they said, "Well, in the fine print, we've said this, this and this. Therefore, we are not liable." It's too bad, but it happened.

Hsu: And Sebastian Thrun was also at CMU.

Reddy: Who?

Hsu: Sebastian Thrun.

Reddy: Yes. Sebastian is another very interesting case. I mean, he came from Germany and I believe he got his Ph.D. from CMU under Tom Mitchell. And he was there on the faculty and worked on a number of things. And he actually was broadly knowledgeable about what Red Whittaker was doing and all the other people were doing in robotics. But he himself was not working on mobile systems at that time. But he knew, though, about them. And what happened was I think about 12 months or 18 months before the Grand Challenge thing, he was hired by Stanford because they were desperate to have someone in robotics. And he went in there and then DARPA announced this Grand Challenge. So he said, "Yeah, we can do that." He understood all the things. He knew also all the post-docs and Ph.D. students that were coming up from CMU, and he recruited some of them. And then he had a huge bunch of brilliant graduates and undergraduates at Stanford working on this project on a crash basis for 12 months. And they won the Urban Grand Challenge at that time. Actually, the Grand Challenge was won and we were celebrating in this building downstairs. Not the Grand Challenge, but that was the year when Cerf and Bob Kahn won the Turing Award. And that celebration was here. And John Hennessey was here. And John Hennessey and I had this bet saying what autonomous vehicles-- you'll be able to get one in ten years or something. And I lost that bet in 2005. And in 2007, when we had this celebration was when Stanford won the Grand Challenge. So John Hennessey said to me, "I didn't think it would happen this fast."

Hsu: So even though he won the… but it was only two years later…

Reddy: He won the bet, too. <overlapping conversation> Usually, most of my bets ultimately come true, whether it is two years later or ten years later. It's just I tend to be optimistic in general, whereas both Gordon [Bell] and John Hennessey are much more pragmatic and realistic of what social systems can do and things like that. He said the surprising thing to me is you still cannot buy a fully autonomous car, vehicle, from anybody. Tesla comes closest, right? And it drives itself, but it's not certified to be driven without a driver in the driver's seat. So that's the problem.

Hsu: Yeah. So 2007 was the Urban Challenge. 2005 was the…

Reddy: No. No, I think it was-- it could be, yeah. 2005-- whenever the year that Cerf and-- Vint Cerf and Bob Kahn won the Turing award was the one that was the Grand Challenge. And then two years later-- that was the one that Sebastian Thrun's car won.

Hsu: Right, yeah. Stanley won the first Grand Challenge, and then...

Reddy: The second one was 2007 urban challenge, with Red Whittaker.

Hsu: Yeah. It's interesting though, that there's this rivalry between basically two people, both came out of CMU, but one had now gone to Stanford.

Reddy: Yeah. No, basically, when Sebastian was at CMU, he was one of the many bright roboticists. When he came to Stanford, he was the only in the major roboticist. And then he had all these other people he could attract, saying, "Let's demonstrate to the world that we can do robotics." So it was partly luck, too. Basically, there was a design problem. There was one piece of hardware that didn't function-- it was supposed to-- the way it was worked. Otherwise, Highlander would have won the thing. And if you ever talk to Red, he will show you the hardware piece that malfunctioned. These things happen, software.

Hsu: How did they feel about competing against each other? Was there any poaching?

Reddy: Yeah, I'm sure there's some amount of-- not animosity, but sour grapes. I should have won it. The same thing happened in 2007. CMU won and Sebastian Thrun came number two or something. These things happen. We are basically-- in retrospect, after 20 or 30 years, nobody will care who won which one, because both of them are just the beginning. But as a historical landmark, it'll be there. But many of the historical landmarks never see the light of day unless somebody digs in, like if they read your oral history of somebody. But otherwise, only a few things, like Wright brothers inventing the plane and human-powered flight, will stand the history of time. And when you look back in computing history, the Internet and the World Wide Web and the smartphone will probably be the three that'll be remembered in history. The rest of the things…

Hsu: Yeah. You mentioned Gordon Bell coming to CMU. What was your relationship like? And did you collaborate at all?

Reddy: We didn't collaborate, but we worked together on projects where he was the prime mover.

Hsu: So he was at CMU from-- what year was it again?

Reddy: I think he came into CMU in '68 or '67. He was there from DEC. And he was promoted from associate professor, full professor in '68 or '69. I came in in '69, and we were overlapped for three or four years. And because of Gordon's connections with Digital Equipment, we were able to get all those computers pretty fast and maybe at a lower price than otherwise. And then he went back to DEC, because they wanted him when they did the VAX computer and others. And again, Gordon had to go back and do the thing. So when Gordon was there at CMU, he had a major impact on all the hardware projects and courses. Among other things, he was able to get Newell, who was in AI, to work with him to write a book on computer architectures. Newell-- the original book was Bell and Newell. Then the next editions were Bell, Newell and Siewiorek. And those books might still be around. If not, I can get you one for the museums, too. And the great thing that they did was took all the different architectures up to that point, like 1970, and then kind of organize them into various things. And they kind of said, “You have to look at every computer from a PMS and IPS-- PMS is processors and memories and connections—interconnections [Switch]. PMS. ISP is Instruction Set Processing. What instructions does it execute, and how does it do it? Those are the two different orthogonal dimensions. So they organized all computers after that point along those dimensions, and explained how they differ from each other. It was kind of a theoretical tour de force in computer architectures up to that point. And so then we were building something called multiprocessor systems, a 16-processor with crossbar switch. These days, they’re all on a single-chip, multi-core processor. So it connected that way. And at that time, C.MMP was using the notation from the book. And it’s C, Computer, dot MMP, Multi-Microprocessor systems. So anyway, Gordon had his impact at that time on the thing. And then only when Hennessy and Patterson’s book came out ten years later, or 15 or 20 years later, that became the standard textbook.

Hsu: Oh. But before that, it had been the Bell and Newell?

Reddy: Yeah. So Bell and Newell and Bell, Newell and Siewiorek. I might even have a PDF version of it.

Hsu: And of course, then Bob Sproull also came to CMU.

Reddy: Yes. There are all these different linkages. Bob was a very interesting person, very important in computer graphics and other things. It turned out Bob Sproull was this student at Stanford when I was at Stanford. So I knew him from that time. And sometime around 1965 or ’66, he left and joined NIH as a fellow. That was at the time of the World War-- Vietnam War and so on. He didn’t want to go fight and he didn’t want to go through the draft thing, so he-- this was an alternative service that was permitted to people, technical people with technical competence. So he was hired in that capacity, and he went and spent a couple of years there. Then, he came back to-- when he finished his Ph.D., came to CMU as a professor, an assistant professor. I don't know.

Hsu: What year was that?

Reddy: I believe he came in ’76 or ’77 or one of the-- ’78. And then we-- in ’80-- in ‘1979 we hired Rick Rashid and Gene Ball. And all of them were involved with the Alto-- Xerox PARC Alto computers, using it and so on?

Hsu: At CMU?

Reddy: No, no. Rick Rashid was at Rochester as a student, and then came to CMU after his Ph.D. Gene Ball came with him. He also had his Ph.D.

Hsu: But Sproull came after he had been at PARC, also.

Reddy: Yeah. The common thread was all of us had some interaction with PARC. And all of us contributed in various ways. And I believe when Jerry Feldman was the other faculty member at Stanford with me in the ’66 to ’69, when Jerry went to Rochester as the [founder of the] Department of Computer

Science. And then Rick Rashid was a student at Stanford, went to become a graduate student at Rochester. So Rick Rashid and company came from there. By that time, Bob Sproull was already at CMU. And in fact, Alan Kay also came for a year, I think.

Hsu: Right.

Reddy: And then Bob stayed. And his wife was a professor in the cognitive science department. Together, they were-- at some point, he was responsible for our computer networking project, project conceptualizing. But then other people came and took it over. So he became-- and then from there, he went to Boston. Either he joined-- Sun Microsystems is it? Some company in Boston. The reason they had to set up a lab in Boston is because his wife was working there or something. So I lost touch-- I haven't lost touch with– I still didn't, but I have lost touch with the career path of what he was following. He and Ivan Sutherland were working closely together. And either Ivan brought Bob Sproull or the other way to CMU. And so Ivan came to CMU, spent a year at the Robotics Institute, built a six-legged robot as an experiment in trying to build something all by himself and something. He was a machine builder at parks and so on. So that was in the early '80s. Okay.

Hsu: Yeah. And you mentioned Rick Rashid. So you've spoken before about how Mach came from, originally, a distributed operating system project?

Reddy: Right. So Rick Rashid came to CMU with an idea of a distributed operating system. How we would build it and so on. That was his project. And at that time, I also had a DARPA grant for distributed sensor networks. If you had a set of sensors that were distributed, how would they cooperate? So I said to Rick, "Before I can do my sensor network, you need to build a distributed operating system in which to connect all these things." So that's how, when we first got a set of workstations, and they were interconnected, and Rick Rashid built the Mach operating system and made it run on a set of PERQ workstations or something. And when Steve Jobs came to CMU, he saw that system and said, "That's what I'm going to adopt." And so he hired one of Rick Rashid's students, Avie Tevanian, to join NeXT and later Apple, to use that core operating system. Because whatever we did was all in the open source domain, so he was able to borrow it from us and then built on top of it, which is now a series of operating systems for iPhones and workstations. So it kind of started at that time, where the Mach operating system went from there to NeXT and then NeXT built on top of it. And then he sold that software to Apple. And that's how Steve Jobs went back to Apple and became kind of a de facto head for a while. And then-- interesting history.

Hsu: Yeah. Earlier, you had mentioned Jaime Carbonell. Can you talk about working with him?

Reddy: Yeah. Jaime is very unique. Unfortunately, he passed away just about the beginning of the pandemic, in 2000 [2020]-- in February 28 of 2000 [2020]. And I remember the date very well because I was supposed to see him a week before. I said, "Look, I'm going to go to California. I'll come back and I'll see you on the 28th or 29th." I came back, and the same day, he died. It's kind of sad, because he was the one that we built the whole language initiative around at CMU. And he was the one that was the first head of the Center for Machine Translation. And that was in 1976. And then-- no, 1986. I'm sorry. And then, in

’93, we said, “It’s not just language and not just translation. It’s a whole set of things, retrieval, summarization, identity detection, *and* translation.” So basically Jaime and I were kind of-- so we formed-- at that time, he came to me and said, “What do we do?” I said, “Let’s change the name.” So we changed it from CMT to LTI. That also happened with no consultation with anybody else. We just changed it. And that was in ’93, I think. And Jaime famously coined a phrase at that time, which we try to make happen even now. He’s no longer here, but I think we understand how to make it happen, but it’s fraught with all kinds of problems I’ve mentioned. We coined the phrase, “AI Bill of Rights.” The goal of AI is to get the right information to the right people in the right time and in the right language at the right level of detail. So it turned out, Jaime kind of said, “That’s our slogan. That’s our motto, vision for the LTI, to get the right information to the right people in the right language at the right time at the right level of detail.” And it turns out, if you try to implement such a system, it’s almost impossible to do. I’ll give you an example. The right information to the right people. So ten years later, we had this Indian Ocean tsunami-- earthquake in 2003 or ’04.

Hsu: ’04, yeah.

Reddy: 2004. And it was the largest earthquake, 9.3 or 9.4. and the waves from that were 100 feet, right? And it kind of went past Sri Lanka and Indian-- south Indian coast. And 300,000 people died. So the idea was, “Can you actually build an AI system to get the right information to the right people? And all the people on the south Indian coast who are going to be impacted by it? And somehow they should be told, every person should be told, that there’s 100-foot wave coming in 15 minutes. And then have them go to some safer, higher elevation as fast as they can run or get on a tree. Or trees might get washed away, too, in that kind of force. So to get-- and not only that, it has to be given in the right time. No point in giving it 20 minutes later or something, because by that time it’ll be too late. And the right level of detail, you can’t go into some scientific explanation of the earthquakes and tsunamis. You just have to say, in Tamil language, which is the language-- so it turns out, to do all of those things, if you imagine what kind of technology must exist, number one, the fishermen must have a smartphone. Not everybody at that time had one. But they could. Even now, you could. The second thing is there must be a global intelligence, all-knowing, all-seeing-- may be not one-- where every person is broadcasting or putting up in a publish/subscribe system the information about where he is and what he does and what language he speaks and all the rest of it. But if he put it up publicly for everybody in the whole world to see, then there’s a huge privacy problem. So the question is, how do you organize the posting of information so that he can get the right warning in the Tamil language in the right level of detail? It turns out to be a very interesting, challenging problem. You have to anonymize the data, but the data must be known, so that if this person has to be notified, they can be notified. So people will say, “How can that be-- there’s no privacy, then, if somebody can track you down and send you this information.” And that’s where the system design becomes a very important issue. You can do it. It’s called differential privacy, where you actually aggregate information for the purpose of who needs to know. And then you say, “These thousand people all have this property. They’re all on the south Indian coast.” Right? And then, an independent system has to simply reach out to all the thousand people and say, “Here’s what’s going to happen.” So if all of that is doable, it requires creating an AI with god-like properties. Omniscient, all-knowing, all-seeing, all-capable. And we can do it, but it requires solving a number of technical problems, which I think will be solved, because the benefits that can be had from such a system are great. Whether it will happen in five years, ten years, I don't know. Or 50 years, I don't know. It depends on, ultimately, people recognizing. It’s like COVID. Once COVID became a major problem for the whole-- globally, you know, the whole world, they had to come up with certain restrictions and regulations which people did not accept or did not like.

Hsu: Could we talk about-- so you started, with Jaime and a couple of others, the Carnegie Group?

Reddy: Yes.

Hsu: An expert systems company?

Reddy: Yeah, we started a company called Carnegie Group during the height of the first AI boom in the ‘80s. Before we started Carnegie Group, at Stanford Feigenbaum and company started IntelliCorp and one or two other companies, all of which were based on this knowledge-based system, rule-based system architectures. And so at CMU, we also were doing the same kinds of things. And it was mainly Jaime and Mark Fox and John McDermott. I was kind of included as the older person who’s kind of involved in the thing. So we formed this group called Carnegie Group to provide AI, rule-based systems and configuration systems for companies. And it turned out, Digital Equipment was one of the first to recognize the use of it. Because they would ship a computer with all the various peripherals-- printers, this, that and the other-- to a customer. And it would arrive there, and in those days it was, like, a million dollars. And one little wire would be missing because somebody made a mistake in the configuration. And so they-- in order-- by the time they ordered it and get it and they would deliver it and find it, whatever it was, it was taking a month. That means they’re not getting paid for a month, and it turned out to be a major problem. So John McDermott built a rule-based system for configuration management. That was one of the early applications of AI. And so based on that strength...

Hsu: Was that called XCON?

Reddy: Yeah, I think it may be XCON. Based on that, Digital Equipment was willing to invest a million dollars for ten percent of the company at that time. And the only deal they requested was they didn’t want to give us cash. They said, “You can have all the million dollars of computing.” Which we needed anyway for all the other things. So that was the beginning of it. Since then, a number of other companies, including Boeing and Texas Instruments and US-- the telecom company, I think. So half a dozen people invested in the company. So we never went to venture capital. The whole thing was funded purely by our customers, potential customers. And the arrangement was that they would not only invest, but they would also agree to provide a million dollars a year of support contract, where they would pay us to support them and their AI activities. And so it was a good thing for them because they didn’t have the expertise. And we were able to provide them, at the same time, they were hoping that if the company ever went public, they would recover all the money they put in from the investment that they made in the company. So anyway, it was an interesting venture capital model at that time. Maybe it may have been done later, but I don't know of any. And that’s how we got the Carnegie Group started.

Hsu: Yeah. So DEC was one of the first clients?

Reddy: Was the first.

Hsu: The first client, and XCON was the major product?

Reddy: Yeah. And I don't know which came first. XCON may have come first, and then DEC invested as a result of that. Whichever way it is, they needed to continuously upgrade it and improve XCON. All of that was done by Carnegie Group. So let me see, except for Jaime, I think Mark Fox is still around. And so is John McDermott. Mark is at Toronto, a distinguished professor. And John is working for some larger company. I forget now which one.

Hsu: But then of course the whole expert systems market crashed.

Reddy: It hasn’t crashed. Even now, people are using expert systems. Corporations need rule-based systems, especially this is true in insurance. Even now, I know that lots of insurance companies will need to either pay off some insurance claim or the initial evaluation, all the things are all rule-based. So there’s a big underground thriving market of rule-based systems. They don't call it AI anymore, and there are specific toolkits and so on, but they are all based on the original idea of rule-based systems.

Hsu: Right. But the company itself...

Reddy: The Carnegie Group was bought by some other company-- I forget now-- after ten years, which then was bought by some other person. So at this point, the identity of Carnegie Group is lost.

Hsu: Okay. So it was sort of acquired and-- okay. So I guess did you all break even in that whole thing?

Reddy: Yeah. We did more than break even. We had a bunch of stock, and I held onto mine. It went up, from six dollars to 35 dollars, came down and came further down. And I sold it at one dollar a share or something, because I don't understand the market. And the people get out at the right time, and I wasn’t even tracking it. I just said, “Let it be.” I did fine. At one point, I was worth, like, 20 million dollars. Ultimately when I sold it, it was probably a million dollars left in it.

Hsu: And could you talk about working with Pradeep Khosla?

Reddy: Yes. Pradeep was one of our star students, who worked with Takeo Kanade for the Ph.D., but he was a student in electrical engineering, because we had arrangements with mechanical, electrical and computer science in the Robotics Institute. So he was working with Kanade on 3D vision systems, I think. And after he finished the Ph.D., he joined the faculty in the electrical engineering department. He rose through the ranks, became a full professor and head of the department at some point. And I think that was around 1997 or so, he was the head of the department. And soon after that, he became the dean. And he was the dean for another five, six years-- five, and left in 2015, I think. Or around that, ’14, to become the chancellor of the UC San Diego. I think he’s still there. He must be in the second term of his thing. I don't know. Normally, they seem to leave after ten years, but we’ll see. It is not yet his ten-year point. I think 2023 or ’24.

Hsu: And you've also had quite a few students, high-profile students, who've gone to Microsoft. So Harry Shum, Hsiao-Wuen Hon, Xuedong Huang, Alex Acero have all ended up at Microsoft. First of all, why has there been such a strong pipeline to Microsoft, and what kind of impact have they made?

Reddy: Yeah. Basically, it was a deliberate decision on the part of Microsoft to build up a major speech research group. So the first person to go was Xuedong Huang. And he was hired away by them in 2000-- I'm trying to remember. 1992 or '93. Maybe '92, I would say. And once he was there, he hired Hsiao-Wuen Hon and Mei-Yuh Hwang and some of the other people, Philip Alleva and Alex Acero. Some of them may have come via Apple. Like Kai-Fu Lee was at Apple, and then he went to Silicon Graphics, I think. But Hon went to Apple because of Kai-Fu Lee, and he was the head-- Hon was the head of Singapore Research Labs for Apple for a number of years. That's where he did the Chinese speech recognition system. And then after he finished the Chinese speech recognition system, he was looking around to see where he wanted to go. And by that time, there was all this uncertainty at Apple, '93, '94. And people were leaving, and Kai-Fu Lee had gone to Silicon Graphics. So I think at that point, Hon came to Microsoft, and he has been there ever since. He is now head of all of research and development, a large development group in China under Hon-- they are all under him. So he's been very influential in building up the technical community in AI and in China. Hon, of course-- not Hon, but Harry Shum had a similar path. He went from here to Apple to Microsoft. And in Microsoft, he did a number of things in vision and graphics. At one point, the SIGGRAPH—at SIGGRAPH Harry Shum's group-- had four or five papers out of 16. It was a huge number of papers from Microsoft on that, together. So he had had a major impact, and goes-- went on. He was number two or number three. Basically, Kai-Fu Lee was the first head of MSRA, Microsoft Research [Asia] in Beijing. And then there was another person, the third head of that was Harry Shum. And he was there for five years or six years and then promoted up to Redmond. And then Hsiao-Wuen Hon became the head after that. He's been there. He's the longest-serving head of MSRA for the last 12, 15 years.

Hsu: Let's see, looking at the time here. So let's go back to talking about knowledge, something we really dove into in the beginning. What does understanding mean, in the context of AI and natural language processing?

Reddy: Yeah. So understanding is a very interesting word. In fact, I remember Herb Simon used to give a whole-hour lecture on the word "understanding." And we may even find some papers he wrote on it somewhere. He was a very prolific writer on these things. And it turns out, if you can answer a question, that could be understanding. Okay? And then you can also say, "If you can summarize something, you are understanding what is being said." Or if there are names of people, places and things-- named entity detection is a major problem in natural language if you want to understand. If you have a diagram, in physics or are trying to explain something, "understanding" turns out to be a word people use to explain the problem you are trying to solve in the context of a diagram. Without that, it's very difficult to express in words, but with the diagram-- so there were a whole host of different meanings of the word "understanding." And each of them would imply-- another word Simon used to say is, "If you can read a book, textbook, and answer the questions at the end of the chapter, then you understood the book." And it turned out in '68, '69 and so on, there were a series of students of Herb Simon's who never finished their Ph.D. because this problem was too hard. Even today, reading a chapter and then answering questions is not solved. The closest it comes to is the work of AI2, Allen Institute for AI, where they solved a corresponding problem. Instead of reading a chapter and answering the questions at the end of the chapter, they said, "If you read a book and can take the final exam, examination..." And they tried to do it for AP and failed miserably and it was not-- and then at that time, I said to them, "See if you can do eighth grade science." And so they did the eighth grade science at the suggestion of Paul Allen, and they got a failing grade, you know, a D. Fifty percent or something, in 2015. They were all kind of very disappointed or discouraged. And when they looked at the errors that the system was making, it was errors of understanding the words.

Hsu: So back to semantics again.

Reddy: Semantics again. It's like-- the best example Jaime used to give me, and I use it all the time, is the word "bank." You could say, is it a river bank or is it a money bank, or is it banking of a plane, or you can bank on me. All kinds of-- there's 17 possible meanings of the word bank. So if I said, "I'll meet you at the bank-- near the river bank," then you're talking about a specific location near the river, the bank of the river. Something like that. So the same word have different meanings. So they discovered all the errors they were making were of this ambiguity resolution. And so they made a series of improvements in their question answering-- understanding of the question so that they could answer it. See, once you understand the question, that means that you are able to create a mathematical formulation. Then they can solve it and give the answer. So the question understanding turned out to be a major problem. And it took them four years to systematically improve. And the most important improvements came from the work done in natural language, which is now the major push by GPT-3 and GPT-1, 2 and so on. Before that, there were things like BERT and RoBERTa and so on. All of them are systems that are both generative and understanding of the context based on what we sometimes call n-gram models. It's not enough to have simple structural n-grams. You need to have common words in paragraphs and so on. So that whole set of advances in natural language, it started with Word2vec things to neural solutions, all of which are now leading to very interesting results. So GPT-3, both generation and analysis is pretty good. And there is some other group, an Israeli company called AI21 which is trying to incorporate more semantics and less data to achieve the same kind of result as GPT-3. So there are very interesting advances happening as we speak in this whole area.

Hsu: Yeah. I mean, GPT-3 is fascinating. It produces results that looks like it understands, but can it really be said to be understanding when it's purely based on statistics?

Reddy: No. It is purely based on statistics, purely based on structural understanding, not because it understands the thing. And there was an interesting example from Professor Joav Shoham from Stanford where it was understanding something correctly. Then they added one more sentence and it completely got screwed up. So from that point of view, they're very brittle, noise-sensitive. And it is purely based on what data they had available at that time, not because they understand the way human beings may understand something, because they have a much broader model of the world. All of it comes down to the word common sense. But common sense is not an adequate word. It doesn't tell you where you go if you want to build common sense. And there are a number of advances that are coming along where you may have to build literally a large number of different kinds of models, each one dealing with one semantic situation. The interesting thing is, human beings understand it. And there are hypotheses saying the number of models you have to build are finite. Maybe a thousand, maybe a million. And once you build all of them, all of those models of the world, then you can interpret anything that comes along based on one of these. Then once we decide that, then you're okay. And when there are ambiguities, those are the raw material for jokes. When you look at a joke, all it is is a misinterpretation of a sequence of words, which also is an interesting possibility, and so people laugh. And so those are the kinds of things, I think, we'll be studying in the next few years. If you can actually laugh at a joke by understanding that there's a different meaning, and therefore it's worthy of a laugh, that kind of an AI system, how often does it laugh? How often is it correct when it laughs? That's a good question.

Hsu: Interesting. So we talked a bit about knowledge representation earlier. What are the advantages and disadvantages of various ways of representing knowledge? And why has that been such a difficult problem?

Reddy: There have been really serious attempts at knowledge representation. And all of them ultimately go back down to some kind of a word net—like semantic network. Or attempts to kind of categorize everything into things that go to further, finer detail, they share all the properties of the super node and things like that.

Hsu: A hierarchy.

Reddy: A hierarchy. And so there have been a number of attempts at knowledge representation. Somehow, that whole area has not taken off. It should take off, but it's probably being swamped by the deep learning advances, which are-- they are able to infer the knowledge. They may not be able to explain it to you, but they are able to infer the knowledge and act on it. And that was-- any attempt to kind of systematically structure it may be doomed to failure, I don't know.

Hsu: What's your view of Doug Lenat's Cyc?

Reddy: Doug Lenat was the first person who tried to capture the common-sense knowledge. And he believed, and I think rightfully so, that he could capture a large number of situations, common sense, which can be then used to-- the problem is, life is too short. At some point, it's like the telephone operator is connecting things. If we had to have operators today to connect all the phone calls, 90 percent of the population of the world would be telephone operators, connecting the other ten percent. And the same is true with many of these situations. Hand-coding of human knowledge and hand-coding of anything is not a scalable option. It cannot be routinely used to-- you can get some specific examples. If you have a specific problem you can solve it. But in general, hand-programming, if you want to think of today's programmer, is doomed to failure, because there is not enough programmers in the world to solve all the problems that we need to solve. So the right solution is for computers to figure out how to write their own programs, which is a special case of deep learning, namely. In deep learning you're looking at all kinds of other data, images and speech and whatever. And if you limit it to only programs, and program-writing programs, then can you predict what the programs are going to be? And what has happened is I think the Open AI group-- there is a group there who is actually attempting to predict, given one line of the program, what other lines will follow. And they seem to be having some success. And they have large enough data from GitHub and so on to predict. But again, like you said, it is not based on true understanding of what the program is trying to do. They're looking at the structural information and saying, "If this, this and this happen, then the following thing must be this, because at least ten other programmers did the same thing. That doesn't mean they understand what it is.

Hsu: So, I mean, you're talking about general intelligence, right?

Reddy: I'm not sure there is such a thing as general intelligence and special intelligence. All I'm saying is if the goal is to predict what is missing, what is going to happen next by using the past, you can do it. And that's what Amazon does when they do recommender systems. And they're not always right. And insofar as the application you're working on doesn't care, it works beautifully. All that Amazon does is recommend, saying "are you interested in any of this?". And there, if you make a mistake, it doesn't matter.

Hsu: I mean, the goal of AGI is to-- I mean, all the deep learning systems are narrow, right? They have to be retrained for a different thing. The goal of AGI is that they won't need to do that. One system can do multiple <overlapping conversation>.

Reddy: Yeah. The problem is, as far as I know, nobody is working on AGI. They all keep talking about it. Until somebody starts working on it, then the question is, "Define what AGI means." And I give the example from Alan Newell. He's saying the following are attributes of an intelligent system. An intelligent system must learn from a large amount of knowledge. It must learn from experience. It must use large amounts of knowledge. It must operate in real time. It must tolerate error and ambiguity, and it must use symbols and abstractions. And, it must communicate with other agents and other human beings using language. Every one of these attributes is hard to do. Many of the AI systems now just only do one thing. They use rules and knowledge, or learn from experience, like deep learning systems. Sometimes they operate in real time, and sometimes they can deal with ambiguity. But they don't have abstraction, symbols and abstractions. That's why you have to keep learning each time. And that's what we mean by-- symbols and abstractions simply use chunking. And finally, the idea of communicating, explaining what you're doing to somebody else, explaining in natural language. And also, they have to be able to deal with error and ambiguity. They all do. It's like the bank problem, you know, the language. And sometimes they make mistakes and sometimes they don't. And so it's not yet-- so to me, an AGI would satisfy all those requirements. Build a system that has all of them. Now, you may have systems that do two or three or some number. Not all of them. And then, the question is, is anybody working on all of them? If you say GPT-3 is a step towards AGI, it may or may not be, until it solves all these problems. In particular, the concept of symbols and abstractions. Does it understand abstraction? For example, AI21, the Israeli company, claims they understand, they can do abstractions. They can summarize text, and that's part of their claim to fame. So I have to wait and see. So just because it does that in that one narrow domain doesn't mean it's AGI. And so there's no clear definition of success or failure of what AGI is. Until we have it, it will be just fluff. People talking through their hat and saying AGI, AGI. Nobody knows what the hell that means. Whereas if I say I have a speech recognition system, when I speak, it understands, either you know it's working or not working. You don't have a clear-cut definition of success or failure for any

AGI that I've seen, other than just people talking. Not only that, the word AGI was coined by somebody who is not-- I think I know the person. But anyway, if you go back and look at the history, none of us were talking about replacing human beings. None of us, McCarthy, nobody, because all of them were saying, "Can we get a computer to do things that human beings do?" It was a scientific curiosity. What does it mean for a machine to do the same thing? Play chess or prove theorems or whatever. And it was not replacing the human being. It was simply a technical, scientific curiosity. And the assumption was, if we could do it, then we would be able to build apps-- tools to enhance the human capability. And that would be, essentially, what we would have with these systems.

Hsu: That's very interesting, because it seems to me that McCarthy, Newell, Simon, all of these early pioneers, I think they made a lot of very optimistic claims in the early days about AI and what it could do.

Reddy: I guarantee you none of them said, "AI is going to replace human beings," in any way, shape or form.

Hsu: But they may have implied it by what they didn't say.

Reddy: No. See, what happens is other people who don't like the phrase "artificial intelligence," like Dreyfus and so on, implied-- artificial intelligence implies that we are replacing all human intelligence. And therefore, by assumption, you're going to replace the human beings. That has never been...

Hsu: But didn't it-- wasn't it useful in terms of drumming up funding to make all these inflated claims?

Reddy: No.

Hsu: For instance, the ten years to chess claim. Wasn't that...

Reddy: No, all they said is in ten years there will be a computer-- will beat a world champion. It took 40 years, but we demonstrated it. And it never said that it's going to replace a human being in the AGI sense. All that it says is there's a special intelligence, a special problem, and we may be able to build it, right? So what Newell's list is, saying intelligence actually includes all the six or seven attributes, and we need to build all-- any one of them we may be able to build. But we can't build any system that we know that does all of them yet. For example, we still don't know anything about building emotional AI. We still don't know anything about AI that learns from experience. We don't know how to build AI that learns from mistakes. All kinds of things we don't know. Emotions and humor and comedy and creativity and intuition, all kinds of words. Just like AGI, creativity and intuition, there's no definition of success or failure. Until we can define very specifically what are the input and output characteristics of a system, and that will measure creativity-- otherwise, it'll always be a moving target. No matter what I do, they'll say, "That's not AGI because it can't do something else." The interesting thing is, most people on the planet cannot do many of those things. It's not as though human beings have AGI. Most human beings don't have AGI. They have some narrow aspect of it.

Hsu: Yeah. But couldn't it be argued that this recognition that intelligence involves emotion or involves embodiment is a later development. In the early days...

Reddy: Nobody talked about it.

Hsu: Yeah. In the early days, it was about theorem proving and chess playing.

Reddy: The interesting thing, if you look at all of them, every problem that people worked on in the '60s and '70s had a very clear criteria for succeeding or failing. Can you make a machine walk? Can you make a machine see? Can you make a machine prove theorems? Can you make a machine play chess? It either did or it did not. And you knew when it failed and why it failed. But creativity and intuition and humor and all of those, emotion, sometimes you may be able to say, but it's not as though you can say, "I can clearly make it fail in the following ways," <inaudible 02:30:59>. And if so, then what do you need to do?

Hsu: What's your-- I mean, clearly you've been critical of Dreyfus' critique. Couldn't it be argued that these critics, Dreyfus and Weizenbaum, have, in some ways, their critiques have pushed the field further?

Reddy: Yes and no. Basically, they'll push the field in a direction that is completely useless. That whole issue of taking the imitation game or Turing test, because there's no clear criteria of success or failure. I might demonstrate that I have a system that will answer all the questions and pass the Turing test, but tomorrow I can come up with some other example where it will fail. And in fact, this is exactly what happened, even in chess. It turns out, all the chess programs that won, won because there was a clear-cut definition of who the win or loss, or draw, of every chess game. And the same system may fail in the next three attempts. And that depends on understanding-- and that's what was happening now. With deep learning, people are discovering by just adding the right kind of noise into the input, the whole system falls apart. And all that it's showing is the structural brittleness of the solutions we are building. Now, you can say, "If you can remove the brittleness, then you have AGI." It's an unending quest, because we don't know. It's like the computer program that never stops. It's the stopping problem. You don't know when it might stop. It might, but we don't have enough lifetimes to find it.

Hsu: Yeah. One of Dreyfus' critiques is about the lack of embodiment, right? But people like Rodney Brooks have built robots more on that embodied model. So couldn't you say that there has been some influence from that direction?

Reddy: I'm not sure. Basically, the people that criticize along various dimensions, they have nothing better to do. None of them have actually built a system that actually makes it work. Feigenbaum might have, when he built a system for the dialogue thing, and then said, "Well, this is just-- look at the inside of it. It's nothing but tricks." And in that sense, all that it has demonstrated-- it is possible to fool human beings by just some simple notation. The famous saying, "You can fool some of the people all of the time and all of the people some of the time, but not all the people all the time." And that's what we are finding.

Hsu: But don't you feel like there-- I mean, these days, with all the ethical implications of AI, don’t you feel that there is a place for people in other disciplines-- social scientists, philosophers, ethicists-- there is a role for people not in the field to be able to critique AI, especially since it’s making such a big impact.

Reddy: Yeah. There’s absolutely a role in the field for the people like the ones you mentioned, because they’re coming at it not by saying, “We’re trying to do AGI,” or something else. They’re saying, “If you do accomplish some desired goal, like a car that drives itself, you may actually be creating a set of problems that have to be also resolved.” And this happened in every generation. When the original cars came along, people didn’t have driver’s licenses. There were no rules of the road. And people drove in any which way and there will be some accidents. And then based on that, they will make some regulation saying, “You must have a driver’s license. Otherwise you’ll be fined or put in jail or something.” And if you-- so those kind of things came afterwards. First, the accidents had happened, then comes the driver’s license. So similarly, what we are finding out is you can build an autonomous car, but the rules of the road of autonomous car has not been set up properly. When the ethicists come in and point out the issues, the government will make appropriate regulations, saying you cannot let loose an autonomous car unless it satisfies the following criteria. It might satisfy it. It still may have other problems we don't know anything about. For example, it is now possible for AI to detect if somebody is drunk or if somebody is falling asleep behind the wheel. And there are some countries, I think, passing a law saying that you must be able to detect it and stop the person. And so if somebody has an accident and it doesn't work, then it is possible that the automotive company that made the thing might also be liable. So what they’ll do is they’ll build up a whole case thing and say, “Our system works perfectly, it’s just that this guy was not drunk in the way he was supposed to be. He is behaving some other way, and that’s not the same problem that the regulation was trying to solve, so we haven’t a clue how to stop it.”

Hsu: I want to go back to Newell and Simon. So did you agree with them on their physical symbol systems hypothesis?

Reddy: Yes. Basically, all that it is is human beings, in order to deal with overwhelming input of the real world, have created abstractions. The best example I know is the creation of a map. If I show you a Google map versus the real satellite image, there’s not any obvious mapping. However, we have abstracted it into some representation that people are able to interpret and deal with the real world. It’s not saying the map is a completely accurate representation of the real world. It is an abstraction or a symbol. And in that sense, human beings have that unique capability. They’ve been doing it for thousands of years. And people, when they had to say, “I saw a lion,” at that point, they were using some symbol systems.

Hsu: But doesn't the physical symbol systems hypothesis also mean that those abstractions are somehow manifested in some physical structure in the brain?

Reddy: No. I’m sorry, the abstraction, yes, in the following sense. I say, “How are you?” and then I show you the word how, are, you, typed in. The mapping between this signal and those symbols is completely arbitrary. Basically, the brain does it, saying, “When you see this word and this image, they’re the same person.” The name of the person and the image are two completely different representations, and they’re linked. In that sense, the physical symbol system says if you're going to be intelligent, you need to have the ability to abstract and link semantically in the head, the name Raj Reddy and the photograph of Raj Reddy at the same time and say they are one and the same thing.

Hsu: But we don't know what those physical structures are, necessarily?

Reddy: The physical structure-- there's a whole area of study called fMRI, right? Where people are studying exactly that, saying exactly what part of the brain lights up when I show you the word Raj Reddy, and what part of the brain lights up when I show you the image of Raj Reddy. And then are there kind of signals from one part of the brain to the other, saying, "I'm seeing this image, and that's the same as this one" probably is happening. We don't know. That structure has been hypothesized, saying there are all these different symbolic components of the same thing, and there are different representations and different things. The name in written form and the name in auditory form and the photograph are all the same thing, right? Now, what that same thing is, is it yet another third object? I don't believe so. It just so happens these three things are linked together when they all light up, that is the same place.

Hsu: Did you have any differences of opinion from Newell and Simon?

Reddy: I don't think so. Basically, there were differences of opinion. They were not serious, basically. Newell had the best example of that in saying, "You and I may not agree on something. When we don't agree, we should agree on an experiment that will prove or disprove the hypothesis. Either I'm right or you're right. We don't know. And let us kind of agree on an experiment." This is where Dreyfus and so on, they said, "No chess program will beat me." After ten years, it happened. So there, it was very clear, the proof that it can happen or not. There may be other things where we don't yet know how to design an experiment to prove or disprove a hypothesis. That doesn't mean it's either right or wrong, it's just that we don't know how to prove it. And in that sense, Newell and I can be very good friends, even if we disagree eventually about some aspect. He says this and I say that. That's what I do these days. Whenever there's a disagreement, I say, "Let's follow the Newell rule. Can we agree on an experiment that'll either succeed or fail? If not, then it's just a mythology. And most of the politics fall into this realm. There are all these different people where they say, "According to me, these facts are wrong. This is what I am going to believe in," misinformation and disinformation and all of that. And there's nothing you can do to change or anything. You just say, "Well, we don't know how to..." because all that they're doing is kind of defeating some other dogma that was being said to them. This has been—this is nothing new now. It has been historically true. There have been cults throughout who would say there is a second coming of Christ and it will happen on such-and-such a day. And such-and-such a day would come and nothing would happen.

Hsu: We've talked a lot about the difference between deep learning and the knowledge-based, rule-based AI. This has kind of been-- there's historically been this tension between the two. And one of the famous examples is the Minsky and Papert book, *Perceptrons*, was perceived as an attack on Frank Rosenblatt's work. Can you speak to how-- was the symbolic AI community just so dominant and overly hostile towards the machine learning...

Reddy: Well, they were not hostile. They simply ignored them, or simply said, “They’re charlatans,” or something.

Hsu: Well, that’s hostility, right? If you define them as way out of the field?

Reddy: At the same time, most of the same people, like me, I didn’t call them charlatans, I said, “I don't think it’s going to work, so why waste your time?” kind of thing. And now...

Hsu: But also funding, though, right?

Reddy: Huh?

Hsu: I mean, DARPA funding was-- that was at stake.

Reddy: Yeah. This happens. Basically, in history there are lots of examples in biology also, for example, where people with good ideas were not funded because the dogma-- the main group said, “That’s a bad idea. We’re not going to fund it.” Later on, they come and say-- these days, that’s the debate between RNA and DNA. For a long time, DNA was it. Now, slowly, people are discovering, RNA is at the beginning of everything, rather than DNA. DNA is a latecomer to the game. And therefore-- and the same thing happened with deep learning. In 2010, they rejected most of the papers in deep learning. 2015, they accepted all the papers in deep learning. Why? Because they see there’s overwhelming evidence. All that it says is most human beings are reasonable. Once they’re presented with overwhelming evidence, they will change their mind. And now, we all believe that all future AI systems will be learned rather than programmed. It’s just that we don't know how to make the learning systems. And that’ll be the research of the next 50 years.

Hsu: Right. But I mean, you’ve mentioned Geoff Hinton before. It’s very interesting that the trajectory of his career. He was at CMU, then he went to Toronto. He was successful in a way by moving outside of the core areas.

Reddy: No, it turns out Geoff was welcomed. He was respected. We didn’t agree with each other, but we respected him. Basically, yeah, I’m glad you're doing that, even though we are not-- I’m not sure it’s going to work. But that’s okay. We need to try different things, and I’m glad you're doing it. He was never-- he never went away because he was not respected. Otherwise, we wouldn’t have hired him. We hired him and promoted him and we would have kept him. But he, given the British background, wanted to be back in a location where--. And also he wanted a place where, I think, he could make the thing. The main thing to remember is he left in 1987 or ’86. It took him 25 years from that date to prove that maybe there is some truth to his approach. And not everybody has that patience and perseverance to stick with something so long, and they give up.

Hsu: I do find it sort of interesting that to a large extent, the connectionist advances were made outside the mainstream of AI. And then only later, when they became successful, then they became the mainstream. But they were peripheral for a long time, right?

Reddy: Yeah. It may be in the sense of accepting papers for publication, that may be true. But the people at CMU, anyway, they're not only just Geoff Hinton, Scott Fahlman was another major connectionist person. He's still there. And we respected him and I said, "Do it and show it to us that your systems are working." It doesn't matter whether we are right or wrong, we just need systems that work and do things. And right now, I find, in speech anyway, the major problem is what we call USSOP. Unrehearsed Spontaneous Speech from Open Population. That is the holy grail. And I said, "It won't happen in my lifetime because it's just too hard." But what we are finding is with deep learning that problem is more or less getting solved. Except when you use Siri or any of those things, it is making all kinds of mistakes. The mistakes are all proper names or other kinds of things, misspelling your name, for example, Hansen. It should have conceptually, from context, known what the right spelling is for your name. But we are not applying all that semantic knowledge, and therefore it's not working as well as it could. Whereas human being would have done it.

Hsu: I do think one of the things that's most, maybe, unique about you and your leadership at CMU is the fact that you've been ecumenical and supportive of all the different approaches. You were a supporter of Geoff Hinton back in the '80s. It doesn't seem to be that that's necessarily true in other centers of AI. I do wonder if that's something...

Reddy: That may be. I don't know about other centers of-- at CMU, that culture is somehow inborn. Newell was like that, Simon was like that. All of them were saying, "I don't know if I'll do something like that, but if you want to do it, okay. Let's see." And I have specific examples with my own students. Both Jim Baker and Kai-Fu Lee are very successful. When they came and wanted to do it, I said, "My god, it won't work. Why are you wasting your time? I know hidden Markov models and…" but I said, "If you want to do it, fine. Let's spend a year, and then when you fail you can come back and cry on my shoulder." But they didn't, right? They succeeded beyond my expectations. I was delighted for them. There was no thing saying, "My god, I was wrong, and therefore I cannot accept them." There are people like that. There's a classic case of an astrophysicist, Eddington in England, who didn't believe in black holes and Chandrasekhar's predictions. And he blackballed him and said, "This is completely stupid. It doesn't work." And it took him another 50 years before he got his Nobel Prize. And these things, they're historically there. That doesn't mean he is right or wrong, it's just that at that point in time, the common wisdom was there were no black holes. And that's what I think we need to kind of say. I'm sure if he was alive today, he would simply say, "Yeah, I was wrong." Or he may not, but whatever. And in general, it's like Newton saying-- Einstein coming and saying, "The laws are not what they are stated to be. There may be special case, but in general you need reformulation of the laws of Newton." And if Newton was alive, he would probably accept it. He would say, "Yeah."

Hsu: Right. That speaks to Thomas Kuhn's Structure of Scientific Revolutions, the paradigm shift, etc., that a lot of it is just generational. The generation has to change or the social agreement has to shift before the majority shifts over.

Reddy: Right. And so basically, at that point, everybody will say, "Yeah, we were wrong." I say I was wrong. And now, I believe there still is-- knowledge is there, right? But before I used to think of knowledge as predicative, so either true or false. Now, I think knowledge is quantum, probabilistic. And it could be almost right or almost wrong or whatever, but all of that is accumulated knowledge. Like I might say tomorrow morning sun will rise for sure. That's a right statement. But sun rises in the east every day all the time, it would be wrong, because three billion years from now, it won't. When we are no longer here.

Hsu: I guess I do have one more thing. Could you recite a Telugu poem for us?

Reddy: Oh yes. Of course. You want a long poem or a short poem? Short poem probably.

Hsu: Short's fine.

Reddy: Sure <foreign language>. It says when you look at two things, camphor and salt, they look the same. But when you taste it, they're different. Very different. And so just like that, in the world there are some wise people and some stupid people, and you can't just judge them by their looks. Something like that.

Hsu: So it's kind of like don't judge a book by its cover kind of thing?

Reddy: Right. I can tell you five other poems if you want. There's another one, which I like. It's a little bit longer. It says <foreign language>. All it says is, if you are an average person, you probably doubt everything and you won't start anything. And if you're kind of a little bit experimental, you might try something, and then when you fail you give up. If you're a hero, even when you have failures, ultimately until you succeed, you keep trying it until you get them. Like Geoff Hinton. Most of us would have given up after one year or five years or ten years. But he stuck with it for 30 years, right? That's wonderful.

END OF THE INTERVIEW