Allen Newell
A. M. Turing Award, 1975

“Desires and Diversions”

Retrospective talk given at CMU
December 4, 1991

Allen Newell: So, I could just hear people talking, coming up in the elevator and in the hallways. When you get to be two to the sixth years, you know, you can't find any good science to talk about, so you've got to talk about the past. But that's not true it turns out, I could have talked to you about Soar, all right?

So, let’s start. This is actually work by Bob Doorenbos and Milind Tambe, and so here we have a program of about 2,000 productions and it runs from zero decision cycles, which is its basic cycle time, on up to 50,000 decision cycles, and it's chunking all the way. And so here is the number of chunks built. And so, it's actually built eight-thousand chunks by now. So, you're looking at one of the largest production systems ever built. And so, the real interesting question is what's true out there? OK. And so, as all of you know, in this school, in this department, the way Soar is built out of a big production system, and thus there is behind Soar a huge set of rules, ten thousand of them by the time you take the two thousand original and eight thousand. There is a big Rete net there, which is a device for trying to execute those rules efficiently. And so, here's a picture for the same thing which shows the growth of nodes in the Rete net. And there's about 20,000 nodes to begin with and we're getting up towards a hundred and thirty thousand, a hundred twenty-five thousand nodes at the end. So, the Rete net is really big. OK. And you don't want to look at these. There is within each within each task that it runs, there are some internal things that get created for support. So, you really want to look only at the envelope, the underneath envelope. OK, so now if you believe everybody, that system should be slowing down like mad, OK, as it gets so big. That's the Steve Minton story, for those of you who remember Steve Minton.

So, here we have the time per decision cycle and seconds spent on a log scale. And in fact, as you look there's essentially no indication of what we call the average growth effect. No indication that things are slowing down, even though we're already up to about 10,000 productions. Actually, we've been tracking this as we're getting big. When you look here, there's absolutely no indication. Now, maybe you can begin to see that we're finally going to get some average growth effect. So, here we have a really big production system, and of course we're just getting started, and we will soon find out what it's like when it’s sort of out to here, or out to here, when we get 20,000 productions.

So, I could have told you about all these things. OK? But when Merrick [Furst] asked me to give this lecture, and I think this was a form letter he sends to everybody, he said, “Why don't you step back? Instead of just talking about your technical thing, why don't you talk about something sort of more general that reflects upon the larger issues?” So, that's what I decided to do. After all, if I look at myself and I measure myself in decision cycles and you believe the Soar Theory of Cognition, then I'm running at about ten to the ninth decision cycles. This thing is only running at ten to the fifth decision… or ten to the fourth decision cycles. So, I've got quite a bit of extra time going than it does. And so I probably am more interesting than it is. *[laughter]*

I think it’ll be quite a while before we get that system out to ten to the ninth decision cycles. Well, it’s chunking six … one chunk every six decision cycles, that linear curve showed that. Extrapolate out, multiply it by ten to the fifth. So, I've got ten thousand times ten to the fifth that’s ten to the eighth chunks. I've got ten to the eighth chunks, not counting my first 18 years when presumably I was learning nothing. All right. *[laughter]*

So, I said I will reflect not just on my career, but on sort of scientific styles, which might be interesting to the group at hand, since most of you are either haven't yet adopted a scientific style, or probably don't like the one that you're in. So, let's talk about scientific style. Styles of scientific lives. There are a bunch of floating maxims around when we do this. The maxim associated with this slide is “To each scientific life, its own style, but each style defines a separate life”. OK. So, I want to just talk about different styles because I only exhibit one of these and there are lots of other styles.

So, the first one is the one I actually associate with algorithm complexity theorists, which I call a “nomadic existence”. Turns out that it's easy to prove theorems and it's very hard to prove theorems that are not almost like the theorems everybody else has proved. So, every so often in complexity theory someone proves a really new interesting theorem which opens up a new area, and then everybody picks up their tents and they all run over into this new area. And they pick up all the interesting nuggets around, and pretty soon you'll look around and there's nothing around except nuggets that are like everybody else's nuggets. And finally, somebody gets a new one. Everyone picks up their tents. OK. And that's the way you live your life. OK. You live your life moving from one theoretical area to another as it opens up.

The second kind of scientific style is sort of a general substantive theme. Gordon Bell actually has a pretty interesting thing, although he's moved a little bit. He really played out the first half of his career understanding what the nature of computer architectures was. That has shifted for the last 15 or 20 years into understanding multiprocessors. He's not focused on any particular thing. There is no particular machine that he wants to build. He wants to understand the space of multiprocessors and how to build effective multiprocessors. That is, in one sense, a kind of a lifetime effort of his. Although when I talked to him recently he was sort of running out of gas and asking what to do next.

There is another style that's quite different, which I've labeled a sequence of strategic objectives. A kind of paradigm example was a scientist I knew about twenty years ago by the name of Werner Reichardt, who at the time I knew him, was studying the control system in insects that governed their flight. His view of the world — he was very articulate — was that you pick a particular scientific idea. That idea takes of the order of five years to build up and get deep enough so you really can do something, and then you pick another one. So, life for Werner is a sequence of five years, nominally five years project. Each one picked by looking at the state of science at the time.

There are in fact — their number is legion — a number of people, and I see lots of them around here, for whom the goal is simply to work on interesting problems. OK. You simply, in fact, if you can work on interesting problems, that's all you ask of each particular day. There's another variation on this, which is not so pleasant, in which you look at whatever seems publishable. You see a little project in life that looks like it's publishable, let's go for that one. OK. And I know a number of people whose scientific lives are a random walk among the publishable materials. *[laughter]* They, of course, shall all be nameless.

The last style I want to discuss is a single ultimate scientific question in which a scientist adopts a real goal out there, a scientific goal. I have three examples. Herb Simon is now famous for the fact that a single scientific goal of understanding how it is that humans, who are governed by bounded rationality, how all of the phenomena surrounding human cognition and decision making, can be understood by that. That in fact has driven his entire scientific career.

I haven't checked this one out with Raj, but my belief about Raj is that he really has speech recognition as his only fundamental scientific goal. OK. Now Raj does lots of different things. He works on multimedia, works on vision, he’s worked on robotics. Only when he turns back to speech do you see sort of fantastic and interesting things happen. And he keeps coming back to it. So, for Raj, I infer — and I repeat, I didn't ask him — I infer that his life is not all those things. It is really speech as the thing that he wants to really see done with his scientific life.

And for myself, I'm again one of these types of characters. My style is to deal with a single problem, namely the nature of the human mind. That is the one problem that I have cared about throughout my scientific career. And it will last me all the way, all the way to the end.

So, let's look at that a little bit. So, my style is the pursuit of a single desire — that's the “desire” of the title — is the nature of the human mind. The maxim is that science is in the details. Now, what is this question? *[0:10:00]* For me, this question is “How can mind occur, human mind occur in the physical universe?” We now know that the world is governed by physics. We now understand the way biology nestles comfortably within that. The issue is, how will mind do that as well? And the second part of this is, this answer must have the details. I've got to know how the gears clank, and how the pistons go, and all the rest of that detail. That's that question’s *[unintelligible]*. And it's not the same as a bunch of other questions.

For instance, it's not the same as “What is the nature of intelligence?”. That is the A.I. question. And although I have been associated with A.I. for a large part of my life, that's not the question that drives me. This is the question that drives me. And they're not the same.

It's not the Simon question, which is essentially — this is kind of an abbreviated version to get it into a single line — “How does bounded rationality explain all of human behavior?” It's not the same question as that. Although, of course, those questions are close enough so that Herb and I have managed a collaboration for 40 years. But in fact it leads me … my question leads me down to worry about the architecture. It leads Herb to worry about other areas of human life, and how do we explain them in terms of bounded rationality.

It is also not the issue of the nature … it is not the issue of the mind-body problem. That's the philosopher's question. Now they may have the same answer. It may be that the answer to my question and the answer to their question is the same. But it turns out, of course, that my question is a simple scientific question. It may be an ultimate one and hard to get out, but it is a scientific question. And the mind-body is not, it's a philosophical question. And in fact, if the mind-body were cast as a scientific question, all the philosophers would do different things than they now do about it. They’d go out in the laboratory, for instance.

But it's also not “What is the nature of the brain?” That's another question which may have exactly the same answer, mostly because there's really only one answer out there, all right? The same answer. But in fact, if this was the question, I would be doing things very differently than the way I'm doing them.

Now, you do need to realize, if you haven't before, that there is this sort of collection of ultimate scientific questions, and if you're lucky enough to get grabbed by one of these, that will just do you for the rest of their life. Why does the universe exist? When did it start? What's the nature of life? Why is evolution so stable? All of these things are questions of a depth about the nature of our universe that they can hold you for an entire life, and you're then just a little ways into them. That's not true. There are, in fact, some of these — “What is the nature of life?” — which have been essentially answered compared to where they were a hundred years ago.

There are some other rather interesting ones in this list, like to see an atom. There were a number of people, not too many, whose real goal in life was to finally see an atom, having been told by the philosophers that atoms and electrons and all the rest of those things were mere creations of the human mind. They weren't really out there. Then what do you want to do? Well, you want to go see one. Because if you see one, then in fact, you know, it's as real as a chair. And that's, of course, happened now, and you probably take it all for granted. So this, in fact, all four of these … Well, Herb's question is very interesting, because nobody ever quite posed the question that way before Herb did, whereas all the rest of these are sort of obvious questions. They're just deep questions.

So, there's the desire. A question that gets asked naturally at that point is, “If that's what scientific life is, when did it start … for me?” OK. So, the maxim here is that the scientific problem chooses you, you don't choose it. You don't go pick a scientific problem and decide that you are going to go and understand the nature of the human mind.

For me, it was certainly not the problem at age 17, where what I wanted to be most was a forest ranger. And as a matter of fact, I spent one summer up in the high Sierras being one of those kinds, except what I actually did was to take gangrenous calve’s livers, frozen, and chop them up, and grind them up so that the trout fingerlings could have something to eat for a summer. Didn't seem like a great career to me.

It certainly was not at age 19, when I wanted to be, I say, a scientist. I was at Bikini at the time — that's the place where the atomic bomb testing was done — and in the company of all kinds of scientists and medical doctors and so forth concerned with this. And I was clearly caught up in the excitement, although I hardly knew what it was to be a scientist. But I certainly wanted to be one of those. In fact, I was probably a good example of “I want to work on interesting problems”, OK? I suppose that's actually a Chinese scientist.

All right. It was not at age 22, when I wanted most of all to be an optical engineer. Now, optical engineer? Who would ever want to be an optical engineer? Well, it turns out as an undergraduate at Stanford I worked on X-ray microscopy. And most of you think of an x-ray as something which goes through a material so you can't build a microscope with it, can't bend the wave. But it turns out that if you take a surface and you shine the x-rays that a glancing angle, you can get enough index of reflection so you can actually focus the x-rays and make an x-ray microscope. And I was all caught up in the engineering and the physics of doing this as an undergraduate for several years. And therefore, I clearly wanted to be an optical engineer.

It hadn't happened yet at age 27, where what I wanted to be was an organizational scientist. Now by this time I actually was a scientist. As John McDermott would say, “I actually are one”. OK. Because in fact, we had started to study large social organizations. These were air defense organizations. This was right in the time of the Cold War and we were studying — there were radar systems all up and down the coast — and we were studying the behavior and performance of those 40-man organizations as they tried to interpret the air picture flowing in as seen through their radars. And I was engaged in organizational experiments, and they were very successful and very interesting. And at that point I clearly wanted to be an organizational scientist.

But it didn't happen then. It actually happened at age 27.7, *[laughter]* on a Friday afternoon — and I can't remember quite which Friday it was — on a Friday afternoon in mid-November. And another fellow by the name of Oliver Selfridge came to RAND — I was at RAND at the time — an event … and he talked about their system. They were producing a system on what was called the memory test computer for Whirlwind, and they were building a pattern recognition system which had several loops of self-adaptation in it. And I listened — And this wasn't a big seminar. He just got in a room with about three or four of us and described this effort for a couple of hours — and I listened to that, and I knew. OK? This is a genuine conversion experience. I walked out of that place. What I knew actually was that computers can do any kind of complex processing, including learning, abstraction, you name it — that that was a sufficient collection of mechanisms there that were sufficiently rich that we were on our way, OK? It was indeed a conversion experience. And I'm an apostate Christian, so I can tell you, I've been through conversion experiences before, and this is one of those.

I went into the office of my colleague, Cliff Shaw. We weren't of course colleagues on A.I. at that time, but we had been working together on the simulations that we were doing for this sort of thing. And I repeated verbatim, practically, what Oliver Selfridge had said in the previous hour. Then I went home that weekend and I designed a couple of whole systems from scratch, and I've never turned back since. OK?

So, that's what I mean by saying that the scientific problem chooses you, you don't choose it. I suppose you can say, well, I certainly was engaged in a hell of a search during all of the early years of my life when I finally came home in that one hour. Now, the insight, of course, was not the nature of the human mind. That was already there from a kind of organizational study. It was that we could understand by mechanistic means how the mind could operate.

Now as I sort of went over this thing an odd insight — insight’s, wrong word — an odd notion, occurred to me. This is November '54. That's only about a year and a half before what is usually considered this great burst of activity: the Logic Theorist, list processing invented and used, GPS, the General Problem Solver, invented and used, the chess program occurred — all in the course of a year. That's only about a year away, a year and a half away from that thing. Absolutely nothing, if you can go back and examine that, presages why this burst of activity should occur. Very compressed in time.

So, there we are. So, there is desire, and we've got it started. Finally got me quitting wandering around and on target.

But then there are diversions, OK? So, the maxim is “Diversions occur”. *[0:20:00]* They really do, and all you can do is to make them count. And then you can salvage whatever is possible for your main goal. So, I want to illustrate that.

So, here's the first diversion. Gordon Bell came to town. He'd gotten tired of DEC, and so he showed up — and he couldn't stand going to M.I.T. — so he showed up here at CMU to be a professor for a while. And so he was writing a book about all his experience in building machines. And he came around and he gave me a chapter. And I read this chapter, and it was so awful you cannot imagine! I don't know how many of you have heard Gordon speak, but Gordon kind of speaks like this all the time. OK. And he writes worse. *[laughter]*

So, he's a new faculty member and you're supposed to be nice to new faculty members. You know, that's kind of the culture. So, I started to edit this manuscript for Gordon to help him out. And now Gordon, in fact, will organize anything that moves slower than he is, OK? And so, first thing I knew I was co-opted as a coauthor, and there I was with a real diversion on my hand because this book was out of control from the beginning. OK? And as you can see, this is essentially — What is that? — six years, six years that this effort went on. So, you've got to make it count. Now, what actually happened was that within that, for entirely bookish reasons, because we couldn't stand the plethora of notations from all the different reprints that we were doing, we invented two notations called PMS and ISP. Most of you know what ISP is: Instruction Set Processor language. That has become in fact a fairly standard term. Probably nobody knows what PMS is anymore, standing for Processing Memory Switch language, a language of computer configurations. But both of these were major scientific contributions. That's what I mean by making it count.

As far as salvage was concerned, I took one big thing out of this what, six, seven years' experience. Namely, I understood what architectures were about. I don't think I understood what architecture is about when I went in, and in fact, the field didn't either. One of the things that book did, was to help pull together a nascent notion within the computer field about architectures, and the space of architectures, and that's really one of the main contributions to the book. But I sort of salvaged this little piece from this huge effort.

All right. So, here's my second diversion, which is Raj Reddy and the Speech Understanding System. That again goes from, what, seven years, 1970 to 76. There was a small part of this which was trying to help the DARPA community, which needed to have a head of a little committee to tell it to do speech understanding research, what it knew it wanted to do with speech understanding research. That's the function of committees in our society, is when everyone knows what they want to do, you create a committee to tell you. And since I was innocent of this whole thing, Larry Roberts sort of tapped me. But the real reason it turned out was that Raj was here. Raj was one of the people who deeply knew he wanted to work on speech recognition research. So, in fact, the diversion was really to help Raj as well. OK? In fact, if I tell you privately, now that the whole thing is over, it was more to help Raj than to help the DARPA community.

So, we ran the speech understanding thing. I ended up being chairman of the steering committee. I ended up writing a goddamn book about the whole thing. On and on it went. It was in some sense, in its own terms, it was indeed a major success. All I got out of it — that's the salvage part — all I got out of it was at least I understood sensory processing a little bit, which I'd never understood before, and that helps me a little bit in understanding the nature of human cognition and so forth. But again, in terms of the maxim, if you're going to go be diverted, make it count. And we made the speech understanding system count.

So, let's go on. Lots of diversions. So, here's the third diversion. This is Stu Card and Tom Moran and the psychology of HCI.[1] Again, lasted for, what, 11 years. All right? One of the points I will make I, I might as well make it now, is these diversions seem to last a long time. And since the point of diversions is don't let them really get in the way, always come back to the main goal, it really pays to have a long life. OK. *[laughter]*

So, the diversion was, Noel, my wife, and I are both creatures of San Francisco, both born and raised in San Francisco, so we're West Coasters. And worse than that, OK, San Francisco at that time was a very attractive place. And Pittsburgh was not so attractive as it is now. So, there sort of was a need to get something on the West Coast that would help us sort of go back there all the time. So, that was the diversion. And so we generated when Xerox PARC came into existence in 1970-71, we generated a consultantship out there and we invented the notion that maybe it was time to apply all that we understood about cognition to some area, like how humans interacted with computers, or how they did programming, or something like that. And then I convinced two of my graduate students, Stu Card and Tom Moran here, that what they wanted most in the world was to go out and join me at Xerox PARC. Well, I stayed here and just consulted out there at first. And we would create this little unit to go build this thing. All right. So, that's all diversion. Now, it turns out that on the user side, we sort of developed strongly and early, 1975-76, the notion of routine cognitive skills. We devoted a class of models called GOMS for characterizing when humans are operating in a routine fashion in a cognitive skill. We helped establish — there wasn't enough room here to say it — but we actually helped establish the whole field of HCI as an endeavor. In fact, the title of our book, which is called "*[The Psychology of]* Human-Computer Interaction", is sort of the first major use of that term as far as I'm aware. So, you know, there we are again. If you're going to be diverted, at least make good use of it.

Now, what did I salvage? I actually salvaged quite a bit from this effort. Bonnie won't probably like my talking about it as salvage, but that's all right. A piece of this, a kind of a little add on piece of this, was the developing of something called the Model Human Processor, which was an attempt to kind of characterize from what psychology knew, cognitive psychology, all the activities that occurred in the human as they were interacting with the machine. And that gave me a view of what it would be to have a unified theory of cognition. So, I had a real activity, a presaging activity, that set me up for understanding that when I came back to that a number of years later. So, I actually salvaged a lot out of that.

Well, there is at least one more diversion. This is ZOG. Don McCracken and George Robertson and a system called ZOG. The diversion is again too great graduate students and a system, which nobody has heard about called LSTAR. We won't come to LSTAR very much, but I'll touch on it … in which, for a number of odd reasons, but mostly because George and Don were really excited, we set out to produce what you would now call a hypermedia system. Now, we didn't invent hypermedia systems, and in fact, there was another system around called the Promise System, which had a network of thirty thousand frames, OK? And we essentially said that's important from a computer science point of view. Let us create a — these guys are a bunch of medical characters — let us create a version like that in the computer science world. And in fact, we actually help them push the whole notion of hypermedia around. We attempted a major application. We did it. It's a little unclear for reasons I won't go into here, whether it was a success or not. This was in fact putting a whole ZOG system, which is this big hypertext network, putting a whole ZOG

[1] human-computer interaction

system on the Carl Vinson carrier to help manage that carrier's operation. Again, in some sense, if you're going to go be diverted, get some use out of it. Get some general use out of it.

What was there to salvage? Turns out there was only one thing to salvage that I could see. The one thing that happened in our getting into that was that a lot of people, two, three, something like that, a lot of people were really convinced that this network of 30,000, which they soon envisioned being three hundred thousand and so on, was essentially an artificial intelligence. Now, they were wrong. They were absolutely wrong. But that was when Bruce Waxman, one of the characters like this who knew about this system, which was out of the computer science world — so its way over on the side in the medical world — came to me, he said, that's where artificial intelligence really is. You ought to go understand it. What I learned out of this is there are no shortcuts to intelligence. Hypermedia has nothing to do with intelligence. It's not the nature of how humans think. And so those are not connected anymore. *[0:30:00]* But that's the one little lesson that I got.

So, what are the points of note? One of the peculiar ones, but it tells you something, is that all the diversions were for social reasons. Not a single diversion was for scientific reasons. Although if you buy the maxim, once you're trapped in a diversion, make it count. OK. That really fits by the way, if I'm devoted to trying to understand the nature of the human mind, what business do I have been getting diverted from scientific reasons? They've got to be some other kind of reason. And they all were. I like to help Raj and do things like that. All right. They all lasted a long time, but I always came back to the basic scientific issue, my basic scientific goal.

All right. Well, there are other things besides diversions. I mean, you know, desires and diversions — the talk is over. We've gotten through the title. That's all we have to do. Not quite. There are some other things. There are failures, for instance. So, let's talk about failures.

So, the maxim here is, “Embrace failure as part of success”. I always like to say, you know, one good failure a week is just bracing and good for you. *[laughter]* Your problem, is to use it to somehow advance towards the main goal. Something I've not always been successful in doing.

So, now I want to tell you about a couple of failures. Here's the first one. It’s a system called Merlin, which I did with Jim Moore. You'll notice a subtle shift. In the previous thing, where I talk about these diversions, the name came first and the project second. Now in these failures, the project comes first and the name comes second, because it's the project that failed, not the people who were on it. All right. So, this was an effort with Jim Moore, a graduate student here, and Richard Young, who's now over in Cambridge, who was here as a research assistant with me. Again, it lasts for what, six or seven years.

There was a kernel idea behind Merlin, which was again the attempt to do the main line, namely that all thought was seeing one situation as some other further specified. So, if you could see the current situation as the goal just further specified in some ways, then you could see how to get to the goal. So, everything was to be built on a generalized form of mapping in which you would sort of map everything into everything else, and the nature of those maps would, in fact, be the flow structure of thought. It was, in fact, a pretty hot idea and it just had a whole cartload of interesting things. For instance, it had frames, schemas, had attached procedures, had general mapping in it, had notions of indefinite context dependence, had automatic compilation. Remember, this is all — Minsky's frame paper is ’75 — this is all several years before Minsky's frame paper. Had all these great things in. The only problem is we couldn't make it work. OK. That is, it failed.

Now, I need to tell you what it means to say what a failure is. I don't mean that we didn't get a program to make it work. I don't mean that we didn't publish a paper on it. Things succeed in computer science when they affect the course of computer science. And if when you look back five years there's no ripple at all from the papers you publish to the systems you built, that's a failure. OK? So, my criteria of failure is the one we use in tenure committees, which is let's just talk about the way this guy has pushed — or gal — has pushed computer science around. Don't look at the papers, don’t look at the systems. Only ask how they've pushed computer science around. So, in fact, Merlin was really well noticed. It was talked about. It was referenced. It was cited. But it turns out that when you're through, it died. No one knows about Merlin now. Its ideas didn't have any issue. Things like the frames and so forth all came around and this was a little input to them, but not the important input. So, that counts as a real failure. Furthermore, we didn't even learn that the idea was wrong. It's a very seductive idea — may actually be right. OK?

I salvaged almost nothing, almost nothing for my main goal — because in fact I couldn't even learn that the idea was wrong — except for one small technical idea which nobody except myself, and you all in a moment, will know about, which is that the way we constructed Merlin, which was a structure which was a mapping of something which, when you tried to map it on something else, would lead to building a further mapping and building a further mapping — these were called beta structures — those were, in fact, local computationally. That is the associated attached procedure to each mapping, which will be the compilation of how to actually make it work, was in fact, the view from that point and never a global view of what the entire cognitive context was. And so, what I learned out of that was that we had in some sense produced the inverse system from what we wanted to produce. What we really wanted was a system with a single mind's eye — you'll all recognize production systems in that — with a single mind's eye that would somehow survey all of this, and not what we had in Merlin, which actually had production systems in, had little production systems all over this big mapping structure. Small but important technical idea. It’s the only thing I took out of it.

So, let's have another failure. This is the biggest failure of all, by the way. This is something called the Instructable Production System Project. It lasted from '76 to '79. A lot of people worked on this: Lanny Forgy, John McDermott, Mike Rychener, John Laird, Paul Rosenbloom, as young graduate students. The basic idea was to grow production systems by external means. So, as opposed to the designer having to know exactly the structure of rules, you’d sit on the outside and you'd loft rules into the system, and it would sort of grow up, OK? We actually talked about it a lot, as if the goal was to obtain a large production system. In those days, that was a thousand, not ten thousand. OK. No one had reached it. No one was above 200. It was, in fact, a total failure. We never got off the ground. One of the ways — we never produced anything — one of the ways of illustrating this in terms of publications is there was a prospective paper and there was a retrospective paper, but there were no papers in between. *[laughter]* It was so bad, it was so bad that only Mike Rychener, whose loyalty to the project was really very deep, was the only one that was willing to go off and write the retrospective paper.

So, a genuine failure. OK. That's what I say. Embrace them, don't hide them. Be happy with them, all right? Now, it turns out that it had an incredible set of uses. So, in fact, the whole OPS series of production systems, OPS standing for Official Production System for the Instructable Production System project, because we all had our own version, was generated by this project. It turns out that as we struggled to find some tasks for which this poor benighted system … Oh, I have to tell you what's wrong with the system. Didn't tell you that. The real problem is, maybe I'll say it later, I can't remember. The real problem is that if you're sitting on the outside, you dump a production system in, you can't ever do anything to evoke that production. The production is up there someplace with its conditions. You don't know what it is because you don't know the productions that are in there, that's the whole point. And you're fishing around trying to find some way to make it fire and you can't do that. So, we could never get, it wasn't quite that bad, but it was almost that bad. We could never get a scheme which let the things sort of turn over so you can shape it. In trying to find some path like this, John McDermott picked up a task that Sam Fuller had at Digital Equipment, which turned out to be this VAX configuration task. And so the R-1 expert system, which many of you will know because it was the first really successful commercial expert system, grew directly out of this project. The notion of universal weak methods, something that John Laird and I did, grew out of this project. And in fact, Mike went and had a fairly large influence around this campus in getting engineering A.I. going in engineering circles. So, the yield from this failure was actually pretty good.

So, what did I want? What did I salvage from this? Because success, that is yield in terms of uses of the failure, sort of helps justify spending all those years, but it doesn't lead to the main line. So, I salvaged essentially several important things. This was the launching pad. This failure was the launching pad for thought. It was out of this that we understood that you had to do something — here it is — that you had to do something with productions as sort of a flat organization. You needed to put another organization on top of it. That organization turned out to be problem spaces. We also understood that you needed default methods so that when the system wasn't dealing with your productions that you threw in there, it would always have something that would keep it turning over. That turned out to be the Universal Weak Method. Both of these turned out to be central in Soar. So, it was in fact, the launching pad.

So, the yield was actually pretty good. The only remark I would make about both of these failures is, again, failures can last a long time. *[0:40:03]* If you look at that, that's a lot of years. If you look at the other one, wherever is the other one, that's even more years. So, again, long life helps.

Now there's just two failures. I don't want you to believe that I only had two failures in my life, so I'm not going to go over them. But there is another list of failures, which I could talk about if we really want to come back and focus on my failures.

But diversions and failures are not the only things to worry about in trying to achieve that scientific goal. There are successes on subproblems, and the fact that these become abandoned opportunities.

So, the maxim is, solve … I mean, “If you're going to go for this goal, then you got to solve whatever problems are in the way, of whatever character”. OK? So, you solve whatever problems must be solved. And if you're successful on those, then your problem is not to be seduced by them. Because these will be big successes.

Let me give you some examples. List processing is one, OK? Now, lists are viewed now as a hot idea. A little old by now, but in those days, nobody believed them, especially the programming systems community. But in fact, they had in it the ideas of list processing as developed early. So, this is back in the 1950s. They not only had lists, they had the concept of dynamic resource allocation. They had the notion of data type. They had the full notion of recursion. They had the notion of generators and working with streams. All of those programming concepts were in here. And in fact, after a few years of working, the notion of list turned out to be very important, as you know, Lisp is important. Big success!

The real issue is, you can't be seduced into now devoting your life to list processing since it's such a success. And I succeeded in this one. In fact, in the whole efflorescence of the literature, of the scientific literature on list programming — mostly built around Lisp, but that's irrelevant here — like the funarg problem and all the rest of those things, I never participated in this at all. I don't think I ever wrote another article on list processing. So, I was not seduced away from my main goal because of the success of list processing.

Let's have another one: hashing. So, it turns out that this is not long lived. This is sort of a month of July in '61. And it turns out that that some place on Long Island, or something like this, I can't remember, Marvin Minsky and I happened to be together sitting down with some obnoxious fellow who kept insisting that the only way you're going to build systems that have the right properties, is to not program them, but to build new hardware systems, OK? And so sort of there and over the dinner play — we didn't invent, cause I'll tell you the rest of the story at the moment — but we sort of used hashing, which was not called hashing at the time, to show that you could do with software direct associative addressing and you didn't have to build associative machines, which is what this guy was plugging. You could do it all with software, OK? This was essentially my contribution to this dinner. It turns out that that hashing had been invented a couple of years before by a bunch of characters working on a disk system called the RAMAC, in which in order to deal with the accessing problem, they invented some clumsy version of hashing. Again, not called hashing. I can't remember what they actually called it. They may have called it scrambled addressing because that's what I called it. And it took that idea in this very special context, generalized it, saw all the implications of it. It was quite clear what all the kinds of things hashing could do for you in the programming world.

But again, I controlled myself. I wrote one unpublished paper on this — this is a historical lie, but it will do for the purpose of this talk — sent it off to Knuth to put in his book, and then went on from there, even though it was clear how important hashing was. And in fact, one could have gone off down that hashing line. Inventing and developing that technology would have been a fairly thoroughly satisfactory, thoroughly satisfactory, research career.

Let me give you another one: protocol analysis. So, this again, this is kind of long-lasting, this sort of goes from '60 to '72, which is a fairly long time. It turns out that one needs as a sub-project to go understand the nature of the mind, an appropriate form of data. And the idea behind protocol analysis is to use the natural language content of people talking about what's going through their mind at the time that they talk. And in the course of working with these early cognitive programs, Herb and I sort of really use that data effectively, and re-establish — it had all been sort of dumped off as the notion of introspection and introspection was unreliable — we sort of re-established this. This a great idea, and has in fact made a very large impact in cognitive psychology and cognitive science.

Again, I did not follow up on that. What was needed at that point was methodological studies, an attempt to turn protocol analysis into a real engine of research with reliabilities and all the rest of that junk. OK? And I just didn't do that. Now, I'd like to take complete credit, but I did in fact right after this period, engage in a period of automatic protocol analysis. That's one of my failures that came to naught. And so, I did not in fact abandon things the way I should have, but allowed myself to be seduced into believing that I ought to continue to spend parts of my life on protocol analysis.

Now, the fact that when I list my vita and all the rest of this stuff I say, oh, what have I done with my life? Well, I've been all involved with list processing and protocol analysis, and those are two big things. That shouldn't confuse you. All right? I mean, that's just that just for PR. OK. What counts is how are you doing with respect to understanding the nature of the mind? And these things are key subproblems that have to be solved. But as they turn out to be successful and these certainly were successful, then you've got to, in some sense, you've got to know to abandon them, give up all the opportunities they offer — I should go talk to Scott Fahlman I'm sure — give up … That's too deep for most of you, except that Scott has balanced his life between neural nets and working on Common Lisp, and he's faced exactly these issues, and he and I ought to talk about that some time.

All right. I got one more success. This is the whole issue of the efficiency of production systems. What are you going to do when you got a hundred thousand productions? How are you going to be able to select those rules in a tenth of a second or a hundredth of a second? Now, in fact, we've been working on this from '76 and we're still working on it. So, this is really a long lived. There are in fact a whole set of important ideas that have developed. This is a little area within the issue in all of parallel processing of the Rete algorithm, fine-grained parallelism, expensive chunks, unique attributes. I haven't abandoned it yet because we haven't gotten the success out of it that we need so that all of our big production systems are working successfully on parallel processors. I hope I have the courage to abandon it when that time comes. So, the points I want to note about this is, success can last a long time too, not just failures. But you should never ultimately be diverted. And then that remark that I made earlier: it does help to have a long life because you're going to have 10-year diversions and 10-year successes on some problems. You've got to have some time to come back and work on the main thing.

So, I'm finally through with all of those, but I need to talk just about success a little bit. So, here is "Successes and their Cultivation". And there are two maxims associated with this. 1) When you make a success, when you get an insight into the essential problem that you're working on, you preserve it. Don't let it go. You work on it and deepen it. In fact, any really deep idea is going to transform itself radically over time and requires the kind of care and tending for that to happen. The reason that's so important is that deep scientific ideas are exceedingly simple and almost everyone else will see them as trivial. OK? Consequently, you'll be the only one to be able to cultivate them for a number of years.

So, here's a list of what I consider to be sort of essential good ideas. Kind of unimportant what the list is: list processing, search, symbols, problem spaces, weak methods, production systems, knowledge level, chunking, impasses, all of which, almost all of which — the story isn't quite the same for each one of them — all of which have this property that they are simple ideas, and that over the years, each one of these becomes deeper and transforms itself. And I'm only going to give you a single example of that.

So, here's the issue of Problem Spaces. I could have picked several other ones, although this was sort of probably the best one to illustrate it. This idea is so trivial as to be uninteresting to almost everybody I know, OK? Which is to say is, it sort of says if you're going to do search you ought to do it in a state space. State spaces, what's the big deal? We all know state spaces are. They're spaces where you have states and transitions between them. That's all is to be said about it, except there are kind of a useful formalism.

So, what is a problem space? A problem spaces is a set of states and a set of operators from state to state, and you have a problem that’s go from some initial state to one of a set of desired states by applying that sequence of operators. All seems fairly straightforward.

But now look what's happened over time. In '57 when this idea first showed up, it was that a search space was the appropriate thing to do heuristic combinatorial search — that is you were just discovering in A.I. that there was a combinatorial search problem, and that you had to kind of modulate that with heuristics, and that's why it wasn't called the problem space quite then. It was simply called the search space. That was the initial idea.

By '65, this idea had transmuted into “That is the space in which *all* problem solving occurs”. That is when you look at any difficult problem solving, it always occurs in some problem space. That wasn't clear up here at all. There were just some problems that were formulated this way. So that's a significant change.

A few years later — I can't quite pin down the date — the idea surfaced, the idea was finally understood that, unlike simple little A.I. programs, real human intelligence has such a huge body of knowledge that its problem is that for any task it wants to do, it must select out a small arena in which to do this task and it must get rid of almost everything else that it wants to do. So, there is a deliberate limitation to an arena. And the problem space is the notion that relates to how the human limits the arena. Whether in fact they used it for search or not, is irrelevant. There's this notion of limiting the arena because a human can do so many other things that it couldn't possibly in some sense consider all of them, or bring all of that knowledge, or ask about all of that knowledge. It’s got to impose. And this is done deliberately, of course, so we can change that arena and work on a problem differently.

By 1979 this notion had gotten enriched because it wasn't just problem-solving — that should have been not the “area”, should have been the “arena” again — in which the problem space was the organization in which all cognitive activity was supposed to occur — was to occur. OK, these are all hypotheses, of course, about the nature of human behavior, hypotheses which most of the rest of the A.I. and cognitive world is not paying much attention to. This concept is sort of getting deeper and deeper as it goes along. So that shift from just problem solving to all cognitive behavior, routine and otherwise, to be accomplished within problem spaces.

It seems incredible now that all the way through this, one only talked about single problem spaces. And so the idea got changed in '83, with essentially the development of impassing and universal subgoaling in Soar, to the fact that one now had multiple problem spaces, and that the relationship between those problem spaces was not implementation, but was a lack of knowledge in this problem space about how to do things would lead to the setting up of another problem space. And so that enriched the notion of the role of problem spaces in cognitive behavior. There was now to be a whole web of them.

What is I think the biggest transformation occurred in '87, in which problem spaces all of a sudden turned out to be a least-commitment program control construct. OK? Let me try and explain that. It turns out that what you want in an intelligent system is not to commit yourself in advance to how you're going to control things, but to wait to the very last minute to assemble the knowledge right then in order to do control. And so, if you separate all of control knowledge from all of the things you can do, the operators, that is a problem space. So here it is having nothing to do with search at all, but being fundamentally a control construct. And the way we arrived at this actually was by observation of Soar in which having been imbued with search all the way through this, we sort of looked at Soar, and Soar most of the time wasn't doing any search. When you ask, well, what was the problem space doing during all of that time, the answer was it was being this control construct of delaying to the moment of action, the assembly of control knowledge that might have been generated just a little bit before, in order to do the control at that point.

But we're not through yet. In 1990, which is sort of — this is actually a little odd in time, but I'll just let it go — the idea that an intelligent system has the problem of how it formulates tasks out of nothing. Where do tasks come from? The programming world says a task comes because a human has a task, and it puts that task into a program. Therefore, don't ask. You know: Child, be quiet! Tasks come from humans! We don't ever have to ask how they come. But if you're talking about what is the nature of human cognition, you must say, how does a human who doesn't have a task, arrive at that task in the first place. And it turns out that the problem space becomes a device for how to formulate tasks out of nothing. In addition to all of these others.

And then the last one has a question mark because we know it's there, but it's totally undeveloped. The problem space is a generalized device for error recovery. Which is to say that there I am in this problem space, which is a search space, and anytime time an error happens, I am already embedded in a larger context in which I can consider alternatives. So that error recovery becomes a very different kind of situation than it is in most programming systems. That is an undeveloped idea, but I assure you it's there.

So here's one simple construct. Yet over time, by nurturing it, it has turned out to be an exceedingly rich construct. And I assume it's going to go on. It's going to go on from there. There's a similar story that can be told for some of the others. I'm not looking at my watch anyway, but I assume I don't have time to talk about those.

So let's talk about choosing the final project. The maxim here is “Choose a final project that will outlast you”. My best example of this is an economist by the name of Fritz Machlup. Fritz was a fairly famous economist at Princeton, had worked in all kinds of areas and in economics, had published in 1962 a book called “Knowledge and its Distribution in the United States”, which was sort of trying to show there really was a knowledge industry, and from an economic point of view you really ought to track the knowledge. You ought to characterize it and do statistical studies on it and so forth.

When Fritz retired from Princeton at the usual retirement time, he said “What I want to do is to write an eight-volume study, which is going to” — it’s like the Knuth issue, right? — "it’s going to have a volume for every chapter of the previous book, which really treats this right”. OK? Now, as time went on, that got expanded to be a ten-volume study and so forth. And when Fritz had a heart attack at age 80, he was up to volume four, and he clearly had succeeded in picking a project which outlasted him. And so, he died with his pen in his hand, which is exactly the right way to die.

So, that's one thing. And the other maxim I would have you attend to is that everything must wait until its time. Science is genuinely the art of the possible, and that's going to be reflected in this list. So, Soar seems like a typical final project, especially as you see it now in which it’s clearly going to outlast me. All right?

Look what has to be done for Soar. A couple of these we've done, but most of them we haven't. It’s got to have the right architecture. It’s got to learn continuously from experience. It’s got to communicate with the external world *easily*. OK, so that's from years from now. It's got to learn continuously from its environment, not just from its internal experience, which is sort of what it does now. It's got to live a long time. OK? Maybe not ten to the ninth decision cycles, but ten to the seventh decision cycles at least. OK? It’s got to — people keep telling it — it’s got to embody all of its tasks simultaneously. You can't build one Soar system for this and one Soar system for that. You've got to put everything in the same thing, because that's what's true of human cognition. It’s got to become very large. It’s got to have ten to the five or ten to the sixth productions or associations or whatever you call them. *[1:00:00]* We're only up to ten thousand. Barely started. It has to have a sense of history and place. In the modern terms, it's got to be situated. OK? It’s got to be a system with a sense of self. It's got to learn from a social community.

All of these things are real problems. All of them, if I take Werner Reichardt’s view, are each worth five years. All right. So, clearly I don't have a chance. OK. Someplace down in here it's all over. So, it will surely outlast me. And so, I have essentially done exactly what Fritz Machlup’s done.

Except that's all not true. Soar is not a final project. It's simply the next project that comes along. In fact, if you look at the time it was chosen, which was '83, I was only fifty-six at the time, much too early to be choosing a final project. The choice, however, was strategic. It was finally coming back and picking as a goal all of the things that were focused on that central desire to understand the nature of the human mind. The fact that you see such a massive commitment to it is really — I won't go through this because we already have kind of — is really the fact that it turns out to be a confluence of so many things that seem right.

All right. So, that's about it. So, I sort of sprinkled throughout this thing a bunch of maxims which I think, at least when I hear myself saying them to my graduate students, I think that they probably characterize my own scientific life. So, the maxim that holds for this last slide is, “To each scientific style, its own maxims”. The maxims that I’ve told you about don't necessarily apply to other scientific styles.

On the other hand, there are in fact a bunch of maxims which didn't quite make it in to that particular way of viewing my scientific life. And I thought that a suitable sort of tattered way of ending this talk would be to just go over some of the other maxims that I have found useful and that, in fact I think, are entirely appropriate for the scientific style in which I've lived, but not necessarily other ones.

So, here's the first one. So, this is a random collection. This is a random residual. First one, if you want to work with the results of Field X, you've got to be a professional in X. To wit: no interdisciplinary activity. And I've got some good examples of this. In cognitive psychology, which was central for me, is central for me in understanding the nature of human cognition, I had to become a cognitive psychologist. I couldn't depend upon my cognitive psychological friends. Now in fact, what that means, as some of the psychologists around here will verify, is psychologists have a way of dealing with issues. They sit there and they sort of shout studies at each other. *[laughter]* And one person says, “The study by so-and-so shows this” and you've got to respond “Oh, but that study is flawed in this particular way”. To which someone else says, “But this study shows this”. And you say, “But I have another study which shows this”. OK. Now, in fact, that's the way cognitive psychologists talk when they're together. If you can't talk that way, if it turns out that you don't know essentially every study that's laid on the conversational table, you are not a cognitive psychologist. So, being a cognitive psychologist means to immerse yourself so deeply in cognitive psychology that you can play that game. And I did, and I do play that game. OK, I got a bunch of interesting anecdotes I won't tell you, but could, about that. *[laughter]*

On the other hand, the linguists play a different game. They play a game of making up counterexamples. They make up these interesting sentences, which they zing at each other across the table. *[laughter]* And you have to deal in real-time online with all those counterexamples by making up other counterexamples of your own or denying, in fact, that's grammatical, or whatever. OK. Very different games from what the cognitive psychologists do.

Now, it turns out — Well, I don't know about the neuropsychologists. They're so embedded in anatomical language that I can't even follow them, it turns out. — but it turns out that this one I did, and these two, I didn't. And so consequently, I have never — I've been crippled this way — I have never in all of my professional life really dealt with linguistics adequately, and I certainly have been unwilling to deal with brain damaged patients and studies and information and so forth — what's now called neurocognitive psychology — simply because, in some sense, I wasn't a professional in either of these areas, and so it wasn't safe for me to go use them. And I think this is true. Don't believe it when they tell you that what you need to do is to collaborate with a psychologist, and he will do this for you, or she will do this for you. They won't and they can't. If you want to work across fields, you become professionals in both fields.

Second one, we are all flawed, OK? We all carry our burden of incapacity. And the real thing about that is you want to know them like the back of your hand, and you don't want to fight them. In my own case, the issue is I'm not a mathematician. Now I know I'm not a mathematician because, in fact, I spent a year as a graduate student at Princeton in Fine Hall, which is the temple of pure mathematics. OK? And I came away from that understanding that I was never going to be a real mathematician. I was never going to prove any of those deep theorems. In fact, I was never going to prove any of those un-deep theorems. And I have designed my scientific life in some sense to simply deal with that. I don't fight it. I know that I'm limited in that respect. And so I work with the things where I'm strong. I understand my flaws and I stay away from them.

Here's a hard one: “There is no substitute for working hard, *very* hard.: OK? I tell this to my graduate students all the time. Just like in professional sports, in professional music, on the concert stage, and so forth, the people who are out there and are going to get the gold, the people who are going to find out what the nature of the mind is, is the people who are not only brilliant, and creative, but work very hard. And you can be as brilliant as you want, as creative as you want, and if you don't work hard, someone's going to get there first. It is a hard story, but it's just very, very true. If you don't want to work that hard, don't go after this scientific style of picking up one of these goals. And you look around and you see that's really true about the scientists who have made those kinds of goals. And it just comes out of this notion of competition at the margin of excellence. That's all that's going on. It's just as true of professional sports as it is science. But unfortunately, it's just as true of science as it is of professional sports.

All right. So, how do new things get started? In all of the experience I've had, they always get started by evolution. That is, we're in a situation like that big failure, and out of that failure comes the ashes, which tells — that Instructable Production System problem — which tells you how to proceed next. That happens all the time in every scientific endeavor. Or they come from chance. I didn't ask Oliver Selfridge to visit that day. Oliver arrived. I didn't know him from Adam. He gives his talk. There I am. I'm hooked for the rest of my life. OK. They do not happen by design. I have designed a large number of starts in my life. Almost all of them have gone down the tubes. So, I designed this is Instructable Production System project. Didn't work. So, the next … so that's all.

So, here's another version that I feel most deeply about. A scientist is a transducer from nature to theory. Don't kid yourself that you have very much to offer. Your problem is to go listen to nature, transduce what it has to say into a theory, but don't believe that the theory is coming from you. It's coming from nature. So, the name of the game is to seek out nature and listen to it. Back in the ’60s, whenever I used to go home kind of depressed because I felt I had learned anything in a week, I would take a new protocol, and I would take an evening, and I would clear the evening away, and I'd spent an evening with that protocol, and I would always find out something about human behavior that I didn't know. OK. Just listen to nature.

Those of you who … to hear me say on Soar “Listen to the architecture” — just a version of that same maxim. This is simply my version of Herb's “Have a secret weapon”. Here the secret weapon is every scientist should have a piece of nature that nobody else knows about that you can listen to that tells you what to do. I don't know what the mathematicians do on that. But then I'm not a mathematician, you see!

All right, so here's the last one: “The science is in the technique. All else is commentary.” All right. All that lives, all that lives in science are those techniques, based on theory of course, but are those techniques which your scientific descendants must use to solve their problem. We know about Hamilton because there's a thing called the Hamiltonian, and there's just a bunch of problems in physics which you wish to formulate with Hamiltonians. *[1:10:00]* OK? And we don't know about a bunch of other characters because their formulations were either too general or they weren't interesting enough. So, it's the techniques that you build that really count, and all the general theories, and the points of view, and the commentaries on them, are all fluff, and they don't really count. OK. So, you’ve got to keep that in mind.

For instance, to pick an example. I had a fellow on the Soar project who went to a meeting recently, and he came back and he said, “Everyone there is all enthused about connectionism”. OK. And he says, “I need to write an article to tell them about these other kinds of architecture so they won't be so enthused about connectionism”. And I said, “Forget it”. I said it a little nicer than that, but I said, “Forget it”. Because, in fact, that paper is just talk. That paper is not science, no matter how good it is. And even if you persuade a few people, the real issue is what techniques will you build that your descendants will have to use to solve their problems?

And in fact, if you look over my scientific career, you will see a whole number of places like the GOMS model, where we — and Problem Behavior Graph which I didn't talk about here — where we deliberately attempted to package our scientific results as techniques that could be used by future generations of scientists. Sometimes we were a success and sometimes we weren't.

I'm not going to talk about this, but I will say what it's all about. This sort of says that if you have an urge to talk generally, sort of do it at invited talks like this one. Or do it on the banquet circuit. Don't kid yourself that these general papers are in fact the stuff out of which your real science is made.

So, that's sort of it. I could have actually started out this whole talk by giving a maxim from my father. My father was a great maxim maker. And I suppose I should say it the way Lyndon Johnson did, you know "what my daddy told me". OK? And what he used to say is — and he said it incessantly — he always said "Keep your eye on the main chance". OK? And if I'd have started out with that maxim, then I probably wouldn't have even had a talk to give because that's sort of the maxim that covers all of the things that I've said in this talk. Thank you. *[applause]*

[1:12:49]

At this point in the video a slide of Newell's entitled "A Deep Idea Transformed" is inserted into the video in three parts over 33 seconds.

[1:13:22]

Raj Reddy: We have time for a few questions.

Allen Newell: Sure, but don't ask me to talk about anybody else's scientific life, except Raj's. I talk about Raj's all the time.

Raj Reddy: Questions? Comments?

Audience Member: So, what are your comments on review papers … philosophizing … *[rest is inaudible]*.

Allen Newell: Don't do them unless [don't] you have something better to do. Look, these are all necessary things to make the scientific enterprise. They are not necessary things to achieve that goal. So, observe, what I said was, that if this is the scientific style, that you're really fixed on achieving some fundamental following, resolving some fundamental scientific question… For Instance, I have never organized scientific meetings. I've almost never been on a program committee. I'm a bad citizen. OK? I believe those things are important. But man, if there's that goal out there, then you've got to be careful because you will just squander your life. Did that *[inaudible]*? *[laughter]*

Audience Member: Oh, I was just curious, in what capacity were you at Fine Hall for a year?

Allen Newell: I was a graduate student in mathematics.

Audience Member: Oh, really?

Allen Newell: Yes. So, I mean, I was there. OK? I really was. I got the citadel. I soaked it up. I left after a year to go out and do game theory at RAND. I knew at that point, deep in my heart of hearts, that I was not a mathematician. But I've got a lot of great stories some time! *[laughter]*

Audience Member: So, until you find your quest, what's the right thing to do? What can you do to increase the chance, increase the random chance you're going to find your quest?

Allen Newell: That's hard because if you observe the nature of my search, it wasn't very directed, right? I don't know. What should I say? Work hard, write review papers because it makes you knowledgeable … *[laughter]* … makes you knowledgeable about lots of things. But the real answer has got to be … has got to be: Try. But be smart enough to get out when it doesn't grab you. Because remember I didn't say you choose it. I said it chooses you. So, you go down and you work on something and you find out after a couple of years that this is perfectly good. I got 10 publications out of it. Everything is going fine. But this isn't going anywhere if I project it for 30 years. I mean, you just look at this and you sort of say "I know that". Even though when I talk to my friends, I can tell them stories. I mean we all are good at building up stories about why what we're doing is really a great path and has all of these things. So, you've got to listen to find out whether you really think that holds. I don't know! Wait for Oliver to come. *[laughter]*

Audience Member: I think in your career you've moved from being a very hands-on … probably coding everything yourself up to a level where working with very large ...

Allen Newell: Right. It's happened to Chuck Noll, too. *[laughter]*

Audience Member: Do you have any comments on how to do that successfully?

Allen Newell: Well, the most common is don't do it! All my friends on the Soar project come around and say, "Heh heh heh heh, when's the last time you coded a Soar system?" And so, there's this little drumbeat that goes on which says, "Are you now chief executive officer or are you chairman of the board?" And in fact, I vowed — this is a terrible phrase by you — I vowed in my youth — all right —I vowed in my youth this would never happen — that now I would be coding Soar systems. I would be down there in the trenches along with everybody else, and it ain't happened. And I feel deeply guilty about this — except that professionally I don't feel guilty about anything — but you know I really don't like it. So, the issue is not to make it successful — I may have done this successfully, I don't know — I shield that from myself — the issue is don't let it happen. But whether you've got an ability that I don't have I don't know.

Audience Member: I have two questions. One's a comment, the other one's a question. The comment is that, I think you sell short your contribution of institution building, committee work, programmatic stuff, politicking for DARPA. And I just I wonder what proportion of your total 90-hour week you think that took?

Allen Newell: Huge!

Audience Member: But you didn't mention that at all!

Allen Newell: Right. Worthless! *[laughter]* Not true! Not true! Not true! I'm sorry, I didn't mean to say it! I really didn't mean to say it! *[laughter]*

Audience Member: If you didn't do it this audience wouldn't be here.

Allen Newell: That's true

Audience Member: This institution probably wouldn't exist the way it does today.

Allen Newell: They tell me that's true. All I can say is, is this was a punctate talk. So, we can only talk about diversions that were punctate. The diversion in my life, which is simply another diversion, to feeling my obligation for the institution, feeling my obligations to ARPA, to sort of continually doing all that activity is kind of the continuous part of the spectrum. OK. It goes on forever. It's gone on all my life. It takes … um … 30 hours a week, leaving the other 50 be to be spent on other things. And most of my life I have sort of felt that that was simply an obligation that one did as a member of the scientific world — and to come back to Frank Ritter's question — and within that one sort of decided whether in fact, you'd spent more time on institution-building and less time of being members of program committees and so forth and sort of make little waves. So, in fact, it has been a big part of my life. And so now I've got to redeem myself. And I'm very proud of it. I'm really proud of this environment. I'm really proud of the way it's developed. But it isn't that goal out there, and somehow you've got to deal with the diversions. And that's really the story of this talk, that there are these diversions, they're real, they're big, they're massive. And you've got to keep living a long time and you've got to keep coming back to work at that goal. *[1:20:00]* Now for the question.

Audience Member: The question was, you talked about technique as living on, aren't there domains in which the discovery itself is what lives on, and not the technique? You actually discover something?

Allen Newell: No. I mean I think there are fields that think there are, but I think in fact, that discoveries that don't develop into technique — oh, this is a terrible thing to say —

Audience Member: Say it!

Allen Newell: I'm going to I just have to re-establish context. If a domain can't get beyond having just discovered, if it can't get to where it can convert those to being things that solve problems that scientists want to solve for their science, then that science is in fact in a pre-paradigm state. It's in a very early stage. Ideas and discoveries in physics and chemistry and biology, all get converted routinely into things you can do later, things that help you to actually do things, like find solutions to diseases, build new devices, or build calculations that you can make.

Now there are fields that don't have that. And in fact, a field like sociology has got itself a real problem since it does have a level of technique, like path analysis and so forth. That technique doesn't go to the heart of the science at all, and it doesn't have the corresponding techniques that come out of the deep conceptual structure. And so it's kind of caught. And so is psychology, with analysis of variance is, at least has been in the past, sort of caught on that same thing, in which the level at which it has technique doesn't penetrate to the level at which it needs its theory. And its theory doesn't translate into calculations and other techniques that it can use. That's my answer. You may wish to dispute.

Audience Member: Yes, the people who have taken to your espousing a single life's goal are …

Allen Newell: I didn't espouse it.

Audience Member: … that you described …

Allen Newell: Yeah, I said that's me. I said there were lots of other things, too.

Audience Member: Right. My question actually, those people were very exceptional people. I guess it's sort of a two-part question. One, to what degree do you think that taking that approach may enable one to be exceptional? And secondly, to what degree is it even appropriate for the vast majority of us *[laughter]* to strive for this kind of goal?

Allen Newell: I'm going to give you the NFL answer to that one. Late in your career, it's different. Late in your career, we all understand we're journeymen. But early in your career, if you're out there playing in the NFL, or if you're in here playing as a young researcher, you'd better believe that in fact, you've got it and you're going to make it. Now, it will turn out that almost nobody will. It will turn out that Soar's all wrong. OK. And consequently, it will turn out that 20 years from now Soar is one of the most forgettable things. And so, the path that I followed did not lead, with all rational analysis of whatever it turns out to be, it did not lead to what I thought was the path. Then I will be another forgettable scientist.

So, I do think, as with any other concentration, I do believe that if you focus, and you're lucky enough to be in the right direction, then you have increased chances. On the other hand, there are increased risks. On the other hand, I never had to face that question. That's part of this issue of saying the thing chooses you. I never decided … but I got to be a little careful, I decided to be an optical engineer with passion, OK, and so in fact, maybe I was just a setup waiting to be imprinted. Turned out Oliver got there first. All the other ones sort of float away. I don't know. But I never did somehow decide to do that as a means to an end. I got committed because that happened to me. And in fact, I believe a number of these other paths ... I had this long conversation over some kind of blue fish in Europe with Werner Reichardt, on this particular day, and he expounded this five-year philosophy. Man, he was extremely articulate about it, about that's the way to live a scientific life. The way he was living it, he had done one 5-year thing that was pretty good. This is what it had got him into the Max Planck Institute at Tübingen. He was now in the second one. He didn't expect to spend his life on this. He expected to somehow skim the cream off of this by getting some deep results. That's a perfectly good way to run a scientific life. I came kind of convinced after listening to this guy over lunch that day.

Yes, Bonnie.

Audience Member: It seems that, from this talk, that you've gotten a tremendous amount of satisfaction from the personal compliments and the compliments of your students, but following on Dave's question, it seems like the organization — I mean, what sort of satisfaction can you get out of turning the world to finding — to understand the mind? I mean, can you get any, or is that just too far away?

Allen Newell: Well, let's see, there's several answers. One of them is there's an omission in my talk. Of all the maxims, I missed one. I don't think it's a maxim. But it is a fact about my scientific life that I have done everything with other people. All of these areas have been with other investigators. Not all of them students. I mean, there's been this lifelong record with Herb, there's been Herb and Cliff Shaw and myself, and there's been other configurations of people. In fact, George Robertson was not my student, Rob Gosworth was not a student after a while, he was a staff member. So, everything has always been cooperative, and in that respect that's right. That got missed because somehow this got too personal to pick that up.

The experiment in Soar, which says there's not only a Soar project, but a Soar community, is clearly an effort to try and move to this other level. Again, not done deliberately. I do very few deliberate things in my life. It has evolved this way. And in fact, that's an ongoing — For those of you who don't know, there is a Soar project which you are aware of because you hear about it every immigration day, and then you forget about it for a while, till the next immigration day — but there is a Soar project that's got quite a few people on it. That project is geographically distributed, not just at where John Laird is at Michigan and Paul Rosenbloom is out at USC. But by now, there's a kind of an international community of about a hundred people, all of whom, in fact, work in a very common way towards trying to push Soar.

I consider this to be a great adventure to see how that goes. My abstract notion is it that's the way science ought to be. In fact, when we look — Bonnie has been involved with this — when we looked at GOMS, and we ask is GOMS a success, the question we ask is not what have we done for GOMS, but whether a bunch of cats out there who we don't know and who don't know us very well, but who are using GOMS, which finally shows that that technique breaks free of its originator. There was a day, for instance, when Game Theory no longer belonged to John Von Neumann. That was a great day for Game Theory, because in some sense, finally, it becomes the property of the scientific community. That's got to happen to any real theory of human cognition. It can no longer be tied even to a community, it's got to be part of *[unintelligible].*

Audience Member: I mean, if you choose a goal that outlasts you, then what do you get your success from, in your own life?

Allen Newell: Well, let's see. There are several answers to that. One of the things is that, I never asked about living a good life. Secondly, in fact, the way you measure whether you've lived a good life, is whether each day has been good. It's like the issue of visiting in Europe. If you go over in Europe and you're in a city for three days, like Paris, and you come out complaining about all the things you didn't do, that's the wrong point of view. If every day you spend in Paris was completely full, you've got nothing to complain about, no matter what else you didn't see in Paris. So, if each day you spend doing science, OK, is full, and interesting, and has its local successes, what more do you want?

You don't want immortality, do you? I mean you certainly — some people think they do — you certainly don't really want to know how it's all going to come out. *[laughter]* That's just a question you shouldn't ask. It does relate to the issue that 20 years from now Soar will be wrong. I'll be over on one side of history. I don't now — I think I did my youthful for a while — I don't now have any notions at all about, or really care at all about whether I'm there for the ages. But by now, of course, this project of mine is me. And so, I don't have to justify it, this project of that scientific goal. That's just me. *[1:30:00]*

Audience Member: Do you consider what you do to be computer science?

Allen Newell: I consider that I have done a hell of a lot of computer science. Is that a good enough answer for you? *[laughter]* In one sense computer science does not usually take itself as understanding the nature of the mind. Maybe it should, but it doesn't. And there's actually lots of reasons for that, because there's that whole technology to build. My goal is therefore not computer science, just like my goal is not A.I. That doesn't keep me from being in a computer science department, providing textbooks on computers, even if they're all diversions. And I certainly have enough both objectivity and immodesty to look at the work on computer structures and saying that was good computer science. That makes real contributions to the area of computer architectures and so forth. And I'm glad Gordon, you know, seduced me into that, and I'm glad I got away, and got back to the other thing. I think that's the way I've got to look at my life. You can't treat the diversions as if they aren't worthwhile. If that's going to happen to you, you're in deep trouble. So, part of this is a glorification of diversions.

Audience Member: You talked about becoming a professional psychologist and not becoming a professional linguist in terms of acquiring or not acquiring a style of argument. That sounds like slightly facetious. Is there a substantial, more substantial …

Allen Newell: Well, I mean the issue with this style of argument is really meant to indicate that there really are levels of professionalism in terms of how deeply you understand the field in the fields own terms, so that you can deal with the problems of the field from the point of view of that field. And this was, in fact, just a kind of a fun way of sort of characterizing different fields. I'm sure I could have done it with respect to linguistics if I had been prepared to make the investment. Somehow, it didn't seem, for reasons that are actually unclear to me, didn't seem pertinent enough, whereas I did make the investment, I made very large investments, in learning cognitive psychology and understanding it. And learning it, of course, is just not reading it. I mean, it is in fact operating within it. It is using it. It is understanding the literature. It is in fact learning to talk. That's just an indicator.

Raj Reddy: Last question. *[Name unintelligible]?*

Audience Member: I wanted to ask you about another one, but it sounds a little bit like a *[unintelligible]*. How would a project every five years that contributes to some central question — has that evolved into a …?

Allen Newell: Sure. I'm trying to think about … I mean, one, let's see — It doesn't fit either of my examples, but it’s a perfectly good example. What I remember of Reichardt was — he didn't have that model — he really was, he really was totally free at each boundary, to look around for what was the next thing. And his rationale for that was in five years the science has moved. So … OK. Now, I can't think of a good example — maybe you’re a good example, I don’t know — I can’t think of a good example — Gordon Bell in my thing isn't a good example because he doesn't actually select out projects in that same kind of focused, multiple trial, I’ve got to do this and then we've got to do this in order to finally get deep enough to make the contribution.

Audience Member: So, do you see your project sort of spiraling towards the central question?

Allen Newell: No, no. No such luck.

Audience Member: A fixed radius around the center? *[laughter]*

Allen Newell: No, just sort of scramble up the hill. So, in fact, and Herb is like this too, I don't know whether Herb is here or not, but Herb is like this too. We get up every morning, and you sort of say what's the next thing to do on this total scientific project. And you go off and do that. And you ignore what you've said in the proposals. *[laughter]* You just totally ignore it and you say “Now is the time we should implement Soar in C”. And you never told anyone you were going to spend a third of your resources implementing Soar in C, but that turns out to be the thing to do, so you go do that. And so you just sort of play that one in real random order, and hope that the funding people never catch up with you. *[laughter]*

Raj Reddy: Thank you, Allen.

Audience: *[applause]*

[1:34:54]