An Interview with

SAUL AMAREL

OH 176

Conducted by Arthur L. Norberg

on

5 October 1989

New Brunswick, NJ

Charles Babbage Institute
Center for the History of Information Processing
University of Minnesota, Minneapolis

Copyright, Charles Babbage Institute

Saul Amarel Interview
5 October 1989

Abstract

Amarel begins the interview with a discussion of his interest in artificial intelligence (AI) and his early research in the field while at Radio Corporation of America. He provides a brief overview AI research at Carnegie-Mellon University and Stanford University in the 1960s and his establishment of the computer science program at Rutgers University in the early 1970s. Amarel also discusses the relationship of AI to computer science. The bulk of the interview concerns the Information Processing Techniques Office's (IPTO) support of research in computer science and artificial intelligence. The primary topics of this discussion are IPTO and Amarel's recruitment as director in 1985, the importance of strategic computing, the creation of the Information Science and Technology Office (ISTO) and the budgeting process for ISTO. Amarel concludes with his thoughts on current directions in AI research.

SAUL AMAREL INTERVIEW

DATE: 5 October 1989 INTERVIEWER: Arthur L. Norberg

LOCATION: New Brunswick, NJ

NORBERG: Saul, I'd like to do three things today, if you don't mind. One of them is to discuss with you your own research program over the past twenty years or so to get a feeling for your place in the field. Secondly, I'd like to ask you to describe to me what you think the principal developments in the field of artificial intelligence have been, that we should focus on in our activities - so as not to try to get you to write a history of the field in thirty minutes. Thirdly, to describe what you know about DARPA's interest in the field over the years, and its non-interests in some areas, and your own time as Director of the IPTO office. So, can we start with the first one and let me ask you to describe your educational background in computer science - where you got your degree; what sorts of things you worked on; how you got into AI; how that field became an interest of yours.

AMAREL: This goes back a long time. My interest in AI started in the early '60s, late '50s. I had a doctorate from Columbia, Electrical Engineering. My main interest was control theory at the time, and then I became interested in learning systems and learning machines - that was very much my initial motivation. That's where control and AI sort of meet, fairly nicely. And learning was very much an important part of the concerns of AI researchers in the early '60s. At that time, I started a group which was called the Computer Theory Research Group at RCA Labs in Princeton, New Jersey. I was there for ten years.

NORBERG: When did you go to RCA?

AMAREL: In '58...'59, early '58, and I was there until '69. Then, from '69...

NORBERG: Let's go back. What sorts of problems did you work on at RCA?

AMAREL: I'll come back. I'll just give you a broad idea about connections. After that, from '69 until now, I've been at

Rutgers. At Rutgers, I founded pretty much the department of computer science and my work continued in AI during that period of time. I've been in various other places on sabbaticals from time to time, but again, the work always was mainly connected to AI research.

Coming back to RCA Labs, my work in AI started pretty much in the beginning of the '60s with support from AFOSR. AFOSR supported that work for the entire decade of the '60s. The emphasis was very much on basic questions of problem solving - always was very much interested in mechanisms for problem solving by machine. I was interested in a variety of methods for problem solving by machine, and I explored all sorts of different areas: such as, theory proving by machine in the propositional calculus; reasoning by machine in situations such as transportation problems. I became extremely interested with representational issues. I found out that a change in representation of a problem or of knowledge used to solve the problem can have enormous impact on the ease with which you can solve the problem. These are questions that involve a shift in point of view. AI still doesn't understand very well this process, because most of the work that is being done in strategies of problem-solving research is being done within a fixed point of view that is formulated by the designer of the programs. I was always interested in seeing to what extent we can mechanize the shift in point of view, but I must say, we still don't quite know how to do it. We have a few nibbles here and there.

NORBERG: What's the basic problem, Saul, that you set out to solve back in the late '60s?

AMAREL: This is very much the basic problem that came out of specific investigations of theorem proving, reasoning about actions, and other things like syntactic analysis of language. So now there are six or seven problem areas, and in all of them I found out that, in addition to strategies of search that were very much the main orientation of AI work in the '60s, the question of formulating the problem in an appropriate way - where appropriate is the big word here - was very important and significant. We did it always ourselves; we put a lot of intelligence ourselves in formulating the problem in such a way that the problem-solving procedure in AI could take and do some work with it to solve it, and it was very clear that whatever the AI problem solving procedure did was very dependent - or the way it did it, how well it did it - it was very dependent on the specific representation of the problem that we gave the machine. So, I was very interested in the question of representation and I did quite a bit of theoretical, conceptual, work along these lines, but I still don't see good solutions to those problems. I'll tell you what I did later on in the '80s.

Another subject I was very much interested in at that time was theory formation by machine. What does it take to form a theory about a set of phenomena, and to what extent we can mechanize that. I did quite a bit of work in an area which has nothing to do with science, but rather it was in programming. Suppose that you're given a specification of a program in terms of several input/output correspondences and a language in terms of which you write a program - how can you generate automatically a program that satisfies all these input/output correspondences? Conceptually the problem is very similar to the one of trying to develop a theory within a system of concepts. Given that you are given some relationships between variables that describe the phenomena, you want to explain those relationships. You want, by being given the left side of the relationship, to find out the right side of the relationship. So, theory formation always has been another extremely interesting part of my interests. I was very interested then in the work done at Stanford around the late '60s on AI and scientific activity, such as the Dendral program, the work of trying to elucidate structures of molecules. I was very close to that work at that time. This would be the RCA time; quite a bit of work was done in AI at that time. But I was not connected to DARPA during that period in terms of support, even though many of my friends in the field were receiving support from DARPA, including many people at CMU whom I knew very well. I was on sabbatical in 1966 at CMU, and this gave much closer rapport to our of work. I gave a seminar on problem representations with Newell and Simon in '66 at CMU, and we had many, many common concerns. They were supported from the beginning by DARPA.

In the '70s, when I started at Rutgers the support of AFOSR stopped and the support by NIH started. I was very much interested in trying to do two things: to have AI explored in the context of something which is real out there, a complex problem, to see whether it's possible to do it, and to focus on reasoning which was not as systematic and clear as I was doing before in theorem proving, language analysis and so on. In this case, the reasoning was supposed to be a combination of systematic reasoning and anecdotal and statistical reasoning, which is the way medical diagnosis reasoning goes. I wanted to see whether these kinds of things could be combined. I wanted to see whether work of the kind that Dendral did in the case of analytical chemistry could be used as a guide in medicine. So, much of the work at that time was on diagnostic problem solving and on reasoning about actions in the context of trying to understand sequences of actions that people perform. This was in the context of a project that we developed at Rutgers called the Research, Resource on Computers in Biomedicine which in time, during the '70s, networked together with the project at Stanford also supported by NIH, that was called the Sumex Project. The idea was not only to have local research in these areas, but also to have interaction through networks and time-sharing and to have collaborators in different parts of the country working on the same program, both in development and - the testing of the program, in accretion of the data of the knowledge bases and so on. That was quite an interesting and heroic time in terms of the development of knowledge-based systems and expert systems. We did quite a bit of work in glaucoma and other diseases. My colleague, Kulikowski, who is now chair of this department, was a key person in the work on glaucoma and other medical areas in the Rutgers Resource.

In the late '70s, even though I continued to be a member of that group and actually was the PI up to '83 (from '71 to '83 with that particular activity), I became more interested in issues of AI in design and engineering. This was the time in which we got some support from DARPA to work in this area, especially in the area of design of VLSI systems. I worked on that with Tom Mitchell, who is now at Carnegie-Mellon, and with Lou Steinberg, who is now director of this project. I came back to some of my fundamental issues in the area of representations in addition to design. The field moved to the point that one could approach some of these issues. Instead of talking about how to choose a representation, I became much more interested in the problem of how to shift from one, which already exists, to another one which is a little better. I did quite a bit of work in the area of automatically devising new kinds of operators - moves, macromoves (I called them) - to speed up computations on the basis of experience in previous computations. Then after that, when I was at DARPA, I became even more interested in problems of design and the possible impact that AI can have on design; and in the last couple of years, I have been putting together plans for having major activities in the area of AI and design in the context of a variety of domains. A couple that interest me particularly are the design of large structures, like ships and airplanes and the design of computer architectures. They are different sorts of things but the complexity of these problems, the representational problems that they represent, and some of the things that they bring forward, which is how to use approximate theories in decision making, are the things, at this point, I want to focus on. In engineering the most important intellectual challenge is to choose the appropriate physical theory that can support your decisions and your projections, your analysis of the situation. You never use a complete theory. You always use parts of the theory, the part that's appropriate for you; you engage in approximations, simplifications. How do people do that? How do they obtain approximations, simplifications? How do they choose the right level of grain, in order to get answers to their questions rather than to have an overall understanding of the field itself. This is an important kind of decision that I think AI can start approaching at this point. It has to do with maintaining, simultaneously, not only a complete theory from principles of a domain, but also various versions of this theory - parts of certain things missing, certain terms dropped and with annotations indicating that this particular approximation or simplification is sufficient in order to handle problems of this or the other class. I'd like to push more now along those lines. There is a lot of connection between these problems and problems of representations in problem solving. So, this is where I am now.

NORBERG: Why wasn't it possible to do this before? You just made a comment that you think it can be done now. Why wasn't it possible to do it before?

AMAREL: Maybe it could have been done. It's not a question of the technology. Technology is not the main issue here. The question is some of the concepts that exist at this point. I mean, some of the concepts that we have now in the area of machine learning are very helpful along these lines, we didn't have them a long time ago. Some of the work that was done in the late '70s by Simon and collaborators on the discovery of theories of physical systems also helped with trying to understand, a little bit, the structure of those theories and what kind of techniques bring about these discoveries. Other work on discovery that was done in the meantime: discovery in mathematics by Doug Lenat, and a couple of students of mine have done work in discovery, one in graph theory, and the other one in the area of Conway numbers. There has been a little more work in areas of discovery of theories of different kinds. There is more work on learning that is related to the questions of discovering something new - a regularity in an environment - that you can use in order to do something with increased problem-solving ability. I think those things make today the situation a little more approachable when you talk about knowledge handling, manipulation of theories, approximations, things like that. They are still very difficult problems, but more approachable today than they were 15 years ago.

NORBERG: Can we go back to RCA? What was RCA's interest in having this sort of activity going on within the company at the time? Let me ask the question differently, I'd like to get a more general ....

AMAREL: RCA was not interested in that.

NORBERG: They were not?

AMAREL: No.

NORBERG: Then what did they hire you to do?

AMAREL: Well, if you looked at RCA Labs at that time, you'd have found a very interesting configuration of research enterprises. The main orientation was to try to be on top of the field in certain areas that would give hardware prominence to RCA. Always, they were interested in hardware problems. But they were also interested in the physics behind the hardware problems. Lots of basic physics has been done there on semi-conductors [interruption], lots of very good physical chemistry was going on, lots of good materials developed. Some interesting system ideas were going on, like content address memories. Jan Rajchman, the director of my lab there, was working on different kinds of memory modalities and technologies and on superconducting switching elements. The research was really out at the boundary of what was possible at that time without a clear understanding of how these things are going to enter into systems that RCA might be building. They started entering the computer field in the '60s, but I didn't see any clear coherence between all of the explorations at RCA Labs and any kind of marketing plan or development plan for the company. On the other hand RCA had such enormous experience in doing things in electronic tubes and components of different kinds, that, in general, they developed an attitude that "it's a good thing to have some fairly forward looking and basic work in some area of physics and chemistry that can allow us, in time, to come up with new components and new circuits and new subsystems." I think this attitude was transferred also in the area of computing - not only in the area of hardware, but also in the areas that are not hardware. I was supposed to develop in the lab the non-hardware component of computers, which was considered to be software and algorithms, and which, for some reason, was labeled theory. It was not something tangible, physical; it was not the hardware stuff. I had a group of about 10 people and we started by working mainly on switching theory. We did quite a bit of work with what today is called neural networks, which is threshold logic. We found several interesting things about the limitations of threshold logic in neural nets, about convergence properties and such. We did a lot of work at that time on flexible logic, which I consider very interesting. A couple of other colleagues and I were the people who started to do some work in Artificial Intelligence. We saw that as a new way of approaching programming at a high level. That is, telling the computer what is the problem rather than how to solve it, and then getting the computer to go on and utilize certain strategies it has for problem solving and get it, by itself, to do its own programming for finding the solution. But this was very visionary stuff at the time. I think in no way it connected to any plans the company might have had to develop those ideas into products - it was very, very far away from commercial application. But still, we were encouraged and tolerated up to '68-'69. At that time, RCA definitely decided to get out of the computer field. It was very clear, the writing was on the wall, that nothing in computer research - not only along the lines studied in my group but in other areas including hardware development - would continue with the same intensity as before; and that's where the work of my group was stopped. I couldn't see, really, any rationale, any clear plan as to how ideas of the sort that we were exploring would find a way into, let's say, products and systems within a horizon of 10 years. Nobody was thinking in these terms.

NORBERG: Who was head of the lab at that time?

AMAREL: Jan Rajchman. Rajchman died just this year. Excellent man. One of the smartest persons I knew. Full of very good ideas.

NORBERG: Were all the group leaders reporting to Rajchman directly?

AMAREL: In computing. He was director of the computer lab. There were other laboratories on television, communications, and so on. But all the groups in the computer area on semi-conductor switching, on memories, on graphics, and on theory, were reporting to Jan who was director. I think, also, probably he had quite a bit to do with the kind of environment that was possible for our work. I am sure that the presence of Rajchman as director of the laboratory, with his broad conception of research, made the work of my group possible. Much of our work was also supported from the outside. The work in AI was done with support from AFOSR. It was published widely. The group received quite a bit of visibility in connection with this work. Also, there was good work in switching theory, which was separate from the work in Artificial Intelligence.

NORBERG: What was the reaction around the lab to this reputation of your group?

AMAREL: It was positive, during that time.

NORBERG: Where was the support for the other groups coming from?

AMAREL: Much of that would be from internal lab funding, but also there was quite a bit of support from OSR, not NSF. There was some Navy support, ONR support; there was no support from DARPA for our group.

In the mid-'60s there was an interesting paper by Maturana Lettvin and McCullough on the visual system of the frog, "What does the frog's eye tell the frog's brain?" There was a lot of interest in RCA labs in this paper and in building a system along the lines suggested by the paper. In many ways, there was a feeling that anything that has to do with understanding visual processes was important to RCA who were very much involved in television and very much involved in processing visual information, in understanding vision, and so on. So we wanted to understand animal vision and not only the physics of vision but, also, the way in which physical processes implemented the information processes of vision. In general, there was a good, positive environment in the labs during that period of time. I don't know if it would have happened without the support from Washington. I still believe the support from Washington was extremely important, the AFOSR support that we received at that time. It stopped in the end of the '60s as a consequence of the Mansfield Amendment.

NORBERG: Mansfield Amendment? Why do you think that would have cut into the activity going on there?

AMAREL: AFOSR cut the support, absolutely, completely cut the support for this kind of basic research at that time. It was finished.

NORBERG: What gave rise to going to Carnegie Mellon in '66? Was that an invitation from them or did you seek the invitation?

AMAREL: No, no an invitation from them. I spent half a year there; I taught a couple of courses, a course that Newell normally taught, and a joint seminar with Newell and Simon. I went to CMU after quite a bit of initial interaction with people like Al Newell and Herb Simon and with some other people there.

NORBERG: Can you tell me about that interaction?

AMAREL: There was a lot of interaction through conferences, meetings, because I was very interested in mainline problem solving ideas starting in '61, I think, '61 - '62, with conferences that ONR, in some cases, organized. There were conferences in Chicago, in Arlington Park, Illinois, in Washington, in L.A. There were many kinds of conferences where we met and discussed things together. There were many visits. I visited Carnegie-Mellon many times. They were invited to come and give talks at RCA labs. So, there was a lot of interaction on a personal level about on-going research in AI at the time.

NORBERG: You just mentioned the Carnegie-Mellon group. Were you interacting with other groups besides the Carnegie-Mellon group?

AMAREL: Mainly the Carnegie-Mellon and the Stanford groups. At Carnegie-Mellon, Newell and Simon were the main two people with whom I interacted, and their students also. Lots of their students are still, at this point, very active in AI: Fikes, Pople, others. Many people who were in my AI class at Carnegie-Mellon are second generation AIers. In Stanford, I interacted mainly with Feigenbaum and Buchanan and also with a group of SRI. The group at SRI included Nilsson, Peter Hart, and Charley Rosen. We were all interested in similar things, mainly in AI approaches to problem solving. Nilsson, actually, in his book on AI and problem solving in '71 very much reflected in his presentation my view of problem solving and the condition of the field as I saw it at the time. But then, his view has shifted. Now, he has different ways of looking at issues in AI. He became more formal. But at that time, we were very, very close. It was in the late '60s and early '70s.

NORBERG: Can you describe your view of the field at that time?

AMAREL: I tried to relate the nature of a problem or a problem class to various methods of problem solving. I was very aware of distinctions between problem classes, which I characterized specifically in terms of the kinds of information, knowledge, that exists in the domain that relates goals to means for achieving them. To the extent that you can reason very clearly and simply from goals to means, going back from goals in trying to choose means, then you have fairly simple problems that I called derivation problems. In cases you cannot easily reason from goals to means, all you can do is to have a generator of possible solutions, that proposes solutions, and a simulator that evaluates a proposed solution and permits you to assess how well you're doing in achieving the goals of the problem. Then you can go back and crank up the generator again and have another proposed solution. The situation is very unwieldy then; it's almost intractable, because all you can do is to generate and test. Usually, you are not very easily guided in the choice of the next proposed solution by the sequence of generation and testing actions that you have taken previously, unless you have some additional information about regularities in the problem domain. All problems of this kind, which I call formation problems, and many theory formation problems are of this kind, are much more difficult than derivation problems, and the approach to their solution requires special kinds of mechanisms. Most of the work that was done in AI, actually, even up to now, has been pretty good in situations where you could reason back from goals in a reasonable way, not in a tight way, but in a reasonable way. You could do good work by reasoning back from goals.

From the beginning of my work in AI, I tried to distinguish between modes of problem solving and characteristics of the problems themselves. Specifically, I made the distinction in terms of the degree to which you can reason back from goals. So, I introduced the notion of a spectrum of problems from derivation problems to formation problems. Along the spectrum, you have different kinds of problem solving procedures that would be appropriate. It matters a lot how you represent a problem. There are cases in which, by learning more about a problem, you can represent it in such a way that it amounts to pushing it over the spectrum towards the derivation end where strong ways of problem solving are possible. You can do that by learning, you can do it in other ways. I have a realistic view of problem solving. There's no single universal mechanism for problem solving, but depending on what you know about a problem, and how you represent what you know about the problem, you may use one of a number of appropriate mechanisms.

NORBERG: You suggested that you have not changed your view very much about the definition of the AI problem and the approach to it, but, that people like Nilsson have changed their view. How is his view now different? Can you tell me?

TAPE 1/SIDE 2

AMAREL: He now looks at rule systems and at properties of rule systems as being an important way of looking at problem-solving situations. Also, he has taken an attitude that AI is applied logic. Logic is not only interesting for AI in the same way that mathematics is interesting for physics because it provides an expressive mechanism and a way of obtaining consequences out of assumptions, but it is by itself a very important thing for AI because it reflects ways of doing things that are intrinsically important for AI. There is a debate in the field about whether this is the case. Nilsson has moved more in the direction of the formalists. I have difficulty with this approach. I still believe that there are lots of unwieldy things going on in AI. We are still in a very early stage of understanding them. We have to have an open mind. We have to allow much more unstructured things going on, lots of experiments that allow for some invention. Everything that could be done within logic would be okay, but it's too stale to restrict the field to that. I would love to use, and I do use logic, but I also use algebra quite a bit in understanding some AI systems. But in the sense of any other scientist who is using a formal system to express things exactly and to be able to obtain consequences out of some assumptions, not because I have a belief that there's something intrinsic in the mathematical system or the logical system that reflects the essence of the discipline. I don't see that calculus reflects the essence of certain parts of physics or that any piece of algebra reflects a particular kind of system that has algebraic models. In the same way, I don't believe that logic in any way has anything intrinsic to do with AI or the cognitive process of problem solving, but I believe it is an extremely useful kind of formal mechanism for describing things, talking about things precisely, and for developing ideas, illuminating things, and in some cases for proving some things because you can work within the rules of deduction for certain logics and so on. So there is a difference, at this point, in the attitude of people in the field - but this is meta-issues - about what is a good way to pursue research in the field. I think people like Newell and Simon remain - and Feigenbaum - in the more unsystematic, let's put it, open, scruffy part of the field. I'm closer to that even though some people believe that I'm more interested in neat, mathematical approaches to subjects. Not really, but I would like to understand things, if I can, mathematically. But if I can't, it's not important. The important thing is to try to get the concepts right, to invent new mechanisms. We are still in the period of invention, so you want to be open-minded about things. So, Nilsson definitely moved to something else; Feigenbaum, no; Buchanan, no; Peter Hart moved from SRI into business and he's definitely taken a pragmatic attitude about what he can do with AI to have an impact on financial systems; Charlie Rosen went to make wine at some point in his life and is not active now professionally in this particular area. I have not had too much contact with MIT over the years.

NORBERG: I was just going to ask you that. Why not?

AMAREL: They were not as interested in problem solving. MIT was not as interested. I mean, at the time that Jim Slagle was there in the early '60s, yes. There was some interesting work - the work he did on problem solving in mathematics, on automating symbolic integration, was seminal and quite relevant. A few other theses along these lines, including a thesis by Jerry Sussman, were focused on problem solving. I think, however, that the main focus of work in problem solving was at Carnegie-Mellon, then at Stanford and SRI. MIT from the beginning gave more thought to issues such as vision. Vision was important from the beginning, as well as robotics. Vision, cum robotics, speech to some extent, then issues of learning. MIT was much more varied, and it covered a very, very broad territory in terms of AI ideas. One of the areas was advice-taking that was introduced initially by McCarthy. Then McCarthy moved to Stanford and continued there his work on advice-taking. Behind the advice-taking effort was a big, interesting question of whether we can we really axiomatize the knowledge that an AI system can use. Can we axiomatize, let's say, simple, common sense knowledge that we use constantly in everyday decisions? Can we use logics, conventional or specialized, to be able to deduce out of axioms of common sense behavior, how to act. McCarthy continued work in this area over the years, and he is still considered among the formalists of the field, even though in addition to that work he has also developed LISP. He has been very, very instrumental in developing the symbol manipulation languages that made AI an experimental discipline. McCarthy moved out of MIT and went to Stanford; his work is at the core of one of the two main strands of AI activity at Stanford. Nilsson's work is close to McCarthy's strand. Feigenbaum's work forms the second strand, together with Buchanan. To sum up, even though I knew people at MIT, particularly Minsky and some of his students, I didn't develop close connections with their work.

NORBERG: Can you characterize the different approaches in some way? I don't want to put words in your mouth, so I don't want to suggest ways of doing it. But is the problem selection at Carnegie-Mellon, and Stanford for that matter, driven by a particular view of what AI is versus the view that might exist at MIT, or might have existed at MIT at the time you're talking about? Newell brings up some of these matters in his historical piece on the field.

AMAREL: Yes, he's much more introspective about that. And he's probably a little bit too neat about that introspection. (laugh) His own view of the world is very significant, because he has very much shaped the orientation of what went in AI over several years at Carnegie-Mellon including the intellectual outlook of many of his students. He has been after a fundamental understanding of the general mechanisms that underlie the information processes in problem solving.

NORBERG: Who is this again?

AMAREL: Newell. Very general. The issue of generality has been very important for him. Even today, he is looking at the issue of general architectures for intelligence. What are the general mechanisms that you can identify that could account for intelligent behavior, for intelligent use of knowledge in the choice of actions. I think the same goals motivate his work within cognitive psychology: What are the fundamental mechanisms of information processing that underlie intelligent problem solving.

NORBERG: Could we characterize that as behaviorist, perhaps? Whereas MIT would be much more technically oriented? Trying to reproduce things like vision.

AMAREL: He is interested in behavior, but it is not just behaviorism because he is very much interested in the mechanism. A behaviorist is not interested as much in the mechanisms that generate behavior; he is interested in characterizing behaviors. I would say he is an architecturalist; he's interested in the mechanisms themselves, but also there is the notion of generality connected to those mechanisms. It's not that you're going to develop opportunistically a mechanism for each problem or classes of problems. I don't think that this is of special interest to him. He's interested in trying to get something which will be, if you wish, the axioms of cognition. I mean the kinds of things that are necessary and small in number, the key principles, that would allow you to understand and explain cognition. I think Simon, perhaps, has a more experimental/exploratory attitude, trying to accommodate a variety of mechanisms and theoretical constructs without having to bind them (at least for the time being) into a single set of basic principles of cognition. He believes that information processing models are the right way of capturing many phenomena of problem solving and thinking of theory formation of perception and of discovery. His program is to probe and to explore lots of problems to see to what extent his general hypothesis is appropriate in their context to find what is doable in various problem classes and to invent various ways in which problems of different kinds can be handled. I think Newell is much more guided by a coherent top level vision about what he's looking for. I think in many ways I'm closer to the view of Simon, which says: I don't necessarily expect to have a single grand architecture of intelligence, that there may be a variety of mechanisms. The situation may be quite messy. I like to understand what kind of problems can be associated with what kind of combinations of mechanisms and why. I'm very concerned about the relationship between the problems and the mechanisms, rather than thinking that all the problems can be approached by a small number of mechanisms. I don't believe it would make much sense especially from the point of view of a machine-oriented person. He wants to develop things that are not only going to be feasible on a machine but are going to work efficiently on a machine. In computer science the important thing is to do things efficiently. We are not logicians; we are not philosophers. We are trying to understand the relationship between a computation and the resources needed for the computation, which is time and space. AI is asking whether and how we can solve some very difficult problems, whose solution requires intelligence by computational methods. Not only in principle, but indeed whether we can solve them within certain bounds of times and space. If you look at many of those problems from the point of view of computational complexity, they are all exponential problems; they are all NP-complete problems, problems that do not have tractable solutions. The question is how can you really handle problems of this kind, or sub problems of this kind, in a tractable manner, in polynomial time? I think it's a big game of how to deflect an exponential into a low degree polynomial. That's the game that AI is playing from the pure computer science point of view. How can you represent the relevant, appropriate knowledge, and how do you distribute it in appropriate places in the computation so that it yields a solution to the problem in a time which is reasonable. The is not whether it is possible, in principle, to solve a problem, to show that a theorem is true or false, or that there is an answer, or not, to a certain question; but you'd also like to do these things within a time that is reasonable and memory that is feasible.

The entire area of AI is pretty much at the forefront of computer science. It's very much part of computer science. I see it as taking the hard problems in computer science and trying to see what you can do with them by using any of the resources that computing can bring to them in terms of machines, programming, interfaces, networks and anything else. But there are challenges there, because the problems are very, very hard, and you want to solve them, again, with limits of time and space.

NORBERG: I'll come back to that, but it seems to me you shifted the answer on me very nicely. It seemed to me you were starting off with a contrast - or at least, I asked you to make a contrast and you started to make one - between Carnegie-Mellon and their approach and MIT. Then, suddenly, everybody was in the same category, and we were talking about AI and computer science. Now, I'm still trying to understand why you were not disposed to be interested more in the MIT group, because I think that might tell me about what they were doing as distinct from what you and Newell and Simon were doing.

AMAREL: I don't think there was a guiding set of principles in MIT....

NORBERG: Then is it just problem selection?

AMAREL: They were also much more interested in problems of vision and control of action. Robotics and vision were important from the beginning. Exploration and learning with Winston. I think the search for interesting, exciting new kinds of computation was always important at MIT - I see that throughout. The paradigms have been changing - for instance Minsky has changed paradigms twice or three times in 20-25 years. He is very good at imagining mechanisms for doing things, not elaborating or going into great depth, but being good in positing them and challenging people to take them and to run with them to see what they can do with them. His theory of frames and then more recently the theory of collaborating systems, and so on. It's a much more diverse group in many ways, much more diverse. I don't see any guiding set of theories and principles and approaches over there. It would be good to talk to some of the people who have a sense of perspective over there. I think Newell may have a good sense of perspective about what MIT has been doing, and I would like to hear his view of that. I don't know to what extent Minsky would have a sense of perspective, but he may. I doubt that McCarthy is interested in such issues of perspective. I think that Pat Winston is a scholar. I think he can make an effort to put things in perspective and see what are the guiding principles and the guiding ideas and goals. I can't find them. I think it's sort of, very broad. They do a great variety of things. They do a variety of things that also give rise to interesting side-effects. In many ways, time-sharing can be seen as a side-effect of work in AI. The development of LISP machines and lots of work in systems and architectures came out from workers in AI. I wouldn't say that the distribution between MIT and CMU approaches is technical vs. behavioral. Certainly not just behavioral in the case of the Carnegie-Mellon group because there's a lot of interesting work with mechanisms there. A person who has good ideas about that and who's interested in the philosophical issue of the distinction of approaches between the Carnegie and the MIT crowds would be Joel Moses, who is now chair of computer science at MIT. I remember once I had a long discussion with him. He has reservations, and he has critical things to say, about Simon's approach to AI; but it's good to see what his view is. (laugh) Because he pretty much represents the MIT view of the world.

NORBERG: I'd like to pick up one more question about this as a possible contrast with some of the other things that we're investigating. That is, in your obtaining money from the Air Force, AFOSR, what was the process of applying for those funds?

AMAREL: Proposal. Very similar to the process with DARPA today because it was not done by a very general process of peer review. I think the people in the AFOSR had quite a bit of say about what it was they wanted done, and usually they got together in national conferences and discussed things with potential grantees. I think it was very close to the style of DARPA later, the early '60s at AFOSR.

NORBERG: So, there weren't prescribed programs that one was applying to. It was just a computing program, and you would submit a proposal for what you wanted to do, and you might get funding and you might not.

AMAREL: I think they were, also, more disposed at the time, as DARPA was, to support specific people rather than specific projects. I remember in the early '60s I was one of the few people who did work on theory formation and on problem representations and I was quite visible, and I think the support was for me as a person. In the same way, other people in CMU were supported as individual researchers; the same thing at MIT.

NORBERG: What sort of evidence, Saul, would you give for that?

AMAREL: I can't really give you, unless I go back to the history of various proposals I submitted, and the give and take that took place in getting things funded.

NORBERG: Well, I guess one piece of evidence would be if you were funded on a continuing basis, or some sort of block grant, and if you changed the research topic that didn't matter. Was that possible or did you get it for projects?

AMAREL: That's exactly what happened. Well, the name was basic work in AI, representations in problem-solving, which was broad enough, and lots of things happened during that time. The choice of specific project areas, such as theorem proving, or reasoning about actions, or parsing, or question answering, was something that I had to do as part of what I thought was appropriate and opportune as things went on. But the titles were quite general, and the support was fairly continuous from, I think, '58... '59 perhaps, to '69, almost 10 years continuously, with renewals, with many conferences that OSR organized, with reports that came out. I think it was a combination of choosing people and developing a portfolio of good pieces of work in the area. It was not, in any way, directed. I don't remember at all that the AFOSR effort at that time was directed.

NORBERG: How did you come to choose to join the faculty at Rutgers?

AMAREL: Oh...things like that are always complex. (Laugh) Just at the time that RCA had decided to stop their computing activity - around that time - it happened that a good friend of mine from Princeton became Dean of a new college in Rutgers, Livingston College. There was no computer science, there were computer services, computer centers that every university had at that time. He felt that in a large university it's always very difficult to start a new discipline. At that time, the mathematics department, as it is today, was dominant, and had their own views about whether or not to do computer science and what kind of computer science to do. He felt by having a completely different college in which he is not going to have a mathematics department.... [You have to understand the complexities of Rutgers at that time. Each one of the colleges was quite independent, and there were faculties that were associated with colleges, but there was a single graduate school where faculties got together. So we had three or four departments of english, for instance, two departments of physics, and so on and so on. There were three departments of mathematics, but there was no department of computer science.] This was a good opportunity to start a computer science department in one of the colleges, and it was an interesting challenge for me to build such a department. So I took three or four of my colleagues from RCA labs at that time. This was one of my conditions, to start with a core rather than with zero. Three or four of us started that year and very quickly moved to eight, nine. We had a Ph.D program in a couple of years. It was an interesting challenge to build something from the beginning in an environment that was positive and supportive without having to fight the various territorial wars with either mathematics or engineering. So I thought it was worthwhile. I did really find out that it was a good decision. It was a very interesting thing to do, in terms of developing and thinking what computer science is all about, developing the curriculum and the Ph.D program and introducing after that a new research effort on computers in biomedicine.

NORBERG: Was this the time that you thought seriously about curriculum development? Because you began publishing on the subject.

AMAREL: Yes it was. I wrote several papers at the time on curriculum development. It was the early '70s at the time we developed the curriculum here and we developed also the Ph.D. program. I also wrote several articles, about "What Computer Science is All About." I was very much concerned about the nature of the discipline. I still am concerned with what it's all about. Today I am especially concerned about the developing separation between computer science and computer engineering, which I think is the wrong way to go. I'd like to bring these two entities together.

NORBERG: Which raises an interesting question for me. This is from one of your publications. I'm interested in the two diagrams that I've just pointed out to you because I'm trying to understand your mode of thinking about computer science at the time. It looks as if figure 1 is basically a description of the field, whereas figure 2. is a description of applications of results from research that goes on in the activities of figure 1. Is that a fair statement?

AMAREL: That's right.

NORBERG: AI appears in the bottom one, not the top one. I guess I found that a little puzzling. Why is AI an application of computer science? Maybe I've caught you off guard by handing you the piece without your having time to review it....

AMAREL: No. In my view, any of those things, including AI, have components that appear here in what is called major areas of theory in computer science. There is a notion of analysis and development of algorithms that you can have in AI as well as in numerical computing. There is a notion of representation in languages, in shells and so on, that you have AI, as well as in other areas of computer science. The high-level languages, the machine-level languages, the architectures and controls schemata of computers are relevant to AI as well as to other areas of computation. I make a distinction in my thinking between systems and problems, or systems and applications. In my view of computer science, I feel that there is a core systems area that includes, if you wish, the hardware and software of computing. There is also another part of the discipline that has to do with the problems, the computational tasks themselves. I do use the same kind of conceptual distinction when I talk about AI problem solving. Even though I am interested in mechanisms (I'm very much interested in mechanisms of problem-solving - they are, if you wish, the systems part of the problem-solving enterprise), I am also interested in the structure of the problems themselves independent of mechanisms. I want to understand what kinds of knowledge they contain and how is this knowledge represented. For the same problem you can have different mechanisms working on it. For any class of problems in some domain, you have concerns with high-level languages, with machine-level languages, with architectures, with hardware, and with related theoretical activities, such as how do you analyze algorithms in the domain. Throughout my thinking about curricula, I never liked the partition of the field into practical work, or let's say professional work, and theoretical work. I think theory should not be separate from the various parts of the domain in the same way that theory is never separate from the exp erimental component of any other scientific enterprise. You have a scientific discipline, and within it you have experimentation and then you have theoretical work. But they are both components of that scientific discipline. In the same way, considering computer science, experimental work and exploratory/inventive work, and theoretical work, all of them are components of the enterprise. You can have as much theoretical work in AI as you can have theoretical work in numerical methods. It's not that you have theoretical work as a separate part of the field.

TAPE 2/SIDE 1

AMAREL: Let me come back to that. Let's take any area, any application area. Let's suppose numerical methods, say solving partial differential equations. Within the discipline, you have activities that have to do with inventing algorithms for how to solve these kinds of problems on certain kinds of machines; you have experimentation with these algorithms and with programs to see what are the error behaviors and so on - issues of accuracy. Then you may have theoretical work about properties and bounds of the algorithms. In the same way in the AI area; you may have inventive work, that says well, this is a possible way of approaching problems of this type. Then, you may have experimental work focusing on the invention. Then, you may have theoretical work. It happens that within AI the exploratory, inventive, and experimental work is much, much more dominant than the theoretical work. We've done much less theoretical work in artificial intelligence than we've done in the area of numerical algorithms, including numerical optimization algorithms. For instance, the entire area of OR and optimization is very nicely overlapped with pieces of computer science. You want to try to find an optimum in a constrained problem. Now, AI is also concerned with finding an optimum in a constrained problem, but usually the problem is not formulated in a mathematical framework such as in OR. No! The problem is presented to you in a much wilder way. Many times the constraints are qualitative; you cannot easily manipulate them within a mathematical framework, so you cannot easily obtain theoretical properties of the optimization procedures. It would be great if in AI also we try to develop theories in the same way as in numerical methods. In some parts of problem solving in AI, we do have that. For instance, for certain algorithms like the A* algorithm for searching, we do have some interesting theories. We have a chance to develop some good theories when we work within fairly formal areas, like theorem proving within natural deduction systems.

Now suppose that you are developing a new kind of machine, which is based on caches, which is based on various new kinds of switching elements, new ways of doing logic. You must have the invention to start with. You must do some experiments and build the thing to see whether it's feasible at all. There is a lot of theory you can do to say whether it's an optimal way of putting together the elements, whether it is the minimal number of elements, whether there are certain things you cannot do at all with the machine, certain things would take a lot of time, that with certain things you cannot use the caches in an effective way, etc. In each area of computer science, I think there is an inventive component, there is an experimental component, there is a theoretical component. I like to view the field also distributed, at least conceptually, between *the mechanisms*, if you wish, the systems, the hardware and the software (which is also a mechanism - it's just clothing the hardware in some sense) and *the problems* themselves for which these mechanisms can be used. I don't think that students in computer science should ignore the problems and only focus on the mechanisms. In that sense, I see the problems of artificial intelligence as a subset of the problems of the field; but I believe they are some of the very demanding problems of the field, and they are part of the grand challenges of this field. By necessity, they challenge all the mechanisms of the field; high performance computing, software, modes of interaction between men and machine, graphics, terminals - all of those things - programming environments. Many of these things were really advanced in the field because people had to experiment with problems of artificial intelligence that were pretty difficult and unwieldy sort of problems, where the issue of program development was much more important than program running - we talked about that at lunch.

NORBERG: But the machine didn't hear any of that. I want to continue to explore this with you, Saul. Is it possible to suggest, even if we stick to only one segment of the computer science community, let's say the AI community, is it possible to suggest that certain groups - let's stick to the United States as well - emphasize the systems element in their thinking about AI problems and some the problem selection area, your figure 2 and, therefore, can I characterize MIT as focusing more on this area and Carnegie-Mellon and Rutgers focusing more on this area?

AMAREL: That's an interesting view, Arthur, I never thought of that. That's an interesting hypothesis, yes.

NORBERG: Is it feasible to pursue further?

AMAREL: I have a feeling that this would be a very interesting, feasible way to pursue further because I would definitely think in some kind of broad way that certainly I would say that MIT has been much more focused on mechanisms and on ways of doing things, means of doing things, machines, software, windows, this, that. I think certainly Carnegie-Mellon and some parts of Stanford have been focusing much more on higher level schemata for problem solving, on problem formulation and on problem classes. Also, the Feigenbaum part of Stanford was much closer to that, and certainly the work at Rutgers is much more close to this point of view. [Taps article copy for emphasis.] My own interest - my personal interest - is to find good ways of relating the two. In that sense, from the point of view of the problem solvers, I find myself asking the question, what kinds of mechanisms or classes of mechanisms are appropriate for what kinds of problems and classes of problems? I don't like to see the one without the other. The two things are important to see together.

NORBERG: Are you aware of other areas of computer science, whether it's in software engineering or architecture or data structures or whatever, are you aware of situations in which there are two camps in those fields as well? Could we make the same statements that I just made about MIT and Carnegie-Mellon for groups in other areas of computer science?

AMAREL: Well, the other area would, for instance, be in, let's say, software technology or in database technology, where you do have to handle both problems and mechanisms. Database technology, you definitely have to handle both problems and problem environments and mechanisms. You do have people who are completely involved in the mechanisms of databases; ways of storing things, retrieving things and so on, versus people who are interested in what is called information systems. These people are interested in broader issues; library systems, the ecology of movement of knowledge in and out of libraries and databases and things like that. I mean there are two different camps of people. If you look at people who work in information systems, people like Don Walker in Bellcore, they are talking about the overall ecology of movement of information and of updating information. They are not thinking in details about coding issues and about protection, and about all the other things that are the mechanisms that make it possible to really use machines to store and retrieve large bodies of data. Now, who are the people who are doing work at the coding level of databases? Not very much in universities, I would say; probably in laboratories of industries and in industries. I'm sure you'll find lots of people in IBM doing that. But you find in universities people who are interested more in the properties of query languages and in the formal properties of deductive data bases, which is more at the level of concern with "problems," and with the theory of database mechanisms. [Figure 2] I have colleagues here at Rutgers who are working on deductive data bases within a logic framework. Now, you present a question which is seen as a statement in logic and then the answer to the question can be seen as a proof of your query within the environment of the database seen as a set of axioms. Can your query be seen as a consequence of the axioms in the database? So, there is quite a bit of work along these lines.
At some point this was regarded as work within AI. It has broken apart. I don't like this fragmentation. I don't like fragmentation in other parts of the discipline. I still consider some of the work on deductive question/answering as a

good part of AI, even if it's a little more streamlined; it is handled now within well-defined frameworks and it's done by people who like much more logical, mathematical thinking. But they are not to be found very frequently in major AI conferences; I think they have their own conferences. In the same way, there is fragmentation in other areas that used to be considered mainline AI, like in pattern recognition or even in some areas of speech. You find separate conferences that concentrate in these areas. Even in knowledge representation there are separate conferences at present; much of the work is concerned with logics for knowledge representation. There is a natural fragmentation of the AI field now into many pieces; that is why I like to see a general AI conference once a year, once in two years, to bring together people from different sub-fields. But, I am rambling and talking about ....

NORBERG: No, you're not. You've hit right on the point, because what I was thinking - I was bringing up the examples just before you were in each case of pattern recognition and speech and thought. But you see, what I'd like to even go further with that and suggest that the people who are working here in the systems level on problems like pattern recognition and speech recognition and so on are perhaps not considered part of the AI community.

AMAREL: In many cases they are not.

NORBERG: They're not.

AMAREL: Speech, at this point, has its own little community. Vision has its own community. There's a lot of effort, in both cases, to try to bring them back into the fold. I was a trustee of the International Joint Conference in AI for six years, from '83 to '89 - this was my last year - and I chaired the international conference in Karlsruhe in '83. In the last few years we got inputs about groups that want to have separate international conferences, be it on vision, or knowledge representation, or in speech, and so on. There is some concern among colleagues - should we allow or not allow that. Frankly, it's not a question of allowing .... (laugh) I mean, science moves its own way, has its own tempo. It's going to happen. The only thing that can be done, in my opinion - I think should be done - is to try to see in what way an international conference or an international kind of organization can accommodate this kind of behavior, this fragmentation and coordinate, if it can, and facilitate rather than segment. So I would, for instance, like to see invited talks of some of the better papers in each one of those specialized conferences presented in the synthetic, global conference. So people who do not go to each of the "parochial" conferences can get a sense of what's going on in some of the other areas. Because, frankly, many times there is a rediscovery of things in one area which is already well known in another area. Even within AI, you do have these kinds of things going on.

NORBERG: To get you to summarize this, then: It seems to me that what you're telling me is that fragmentation is a factor in computer science, period. It's not just in AI.

AMAREL: All over the place. In AI as well as in other areas. Sure.

NORBERG: All right. That's important. Would you like to take a break? [Interruption] I want to raise a new issue.

AMAREL: We've talked enough, now, about my own stuff. Let's talk about computer science.

NORBERG: That's right. Can we go back to the discussion we were having over lunch? I would like to begin it here, for the tape machine, by asking you if you would be kind enough to characterize for me what you see as the DARPA program in IPTO over the years. How would you characterize that program? What would you describe as its important aspects? What were its successes in your view? What were things that were less successful? If you can tell me why some were successful and why some were not, that would be helpful. Start anywhere.

AMAREL: Well, what we did at lunch was not as much characterizing the IPTO program in general, but we tried to see how the work in AI and especially some work in interactive systems in networking and in time-sharing relates to a major vision or conception or goal that, if you wish, drives IPTO over the years, especially in the '60s and '70s. My own view is that things like the principle of interaction and the principle of easy-access to computing are not the basic guiding principle, but rather they are derivatives; both of them are derivative notions. I would like to formulate a more fundamental guiding principle. I think it is the idea of being able to achieve an augmentation of human decision-making with computers. We would like to develop dialogue, collaboration, interaction between man and machine that is going to assist a person in his decisions, in making better decisions, faster decisions, than he can do by himself. In order to be able to do that, we have to develop the technology of networking and interaction and systems like time-sharing, and, of course, the kinds of hardware and software architectures that are needed to enable communication between man and computers in a graceful and easy way. Maybe as part of that, you can see the interest in speech because that's a good way of interacting in a natural way with a machine. In some cases, vision could be viewed as part of that, too, because in some situations you want to show something to a machine, and the machine should be able to take it in, and you can talk and interact with the machine about that object, which you can see and the machine can also see. So, if you have the ability to see together with the machine something and talk about it, you have a marvelous way of having an assistant with whom you can discuss a case and come up with a decision.

But in a much more fundamental way, in order to be able to get augmentation of human decision-making with the help of computers, you want to understand the processes of decision-making better. You want to understand fundamentally the information processing that goes on in decision-making and in problem-solving. That's what I think is at the heart of AI. You want to understand these kinds of processes; you want to find out how you can mechanize some of these processes by machine, not necessarily the same way they are done within the human being, but in a way that is going to be computationally effective. So, if you have a command/control situation, lots of information is coming in, different modalities, some signals, some intelligence, some other things coming in, you want to be able to understand all these things that are coming in by machine. You want the machine to be able to understand them, you want to have machines summarize those things in different ways according to the interest of the person with whom the machine is working. You want to be able to represent those things inside machines in a way that they could be retrieved at later times and analyzed and so on. At the same time, you would like to have the machine provide a certain amount of advice about what the machine would do, would suggest doing, in situations like that, given that there are certain options in the way the problem can be approached.

The person still would be in control, but the important thing is to have the benefit of having an intelligent mechanical assistant that would handle information, maybe sometimes in a way that a human cannot do because of volume or time pressures, and reduce the saturation that comes with a human being in these conditions. Not only do that, and then present the stuff summarized to a person, but go the step further and try to see in what way this information could be used together with some other predigested information about strategy and ways of doing things and tactics and give some advice about what to do. That is the function in command/control of the staff member to a commanding officer. Can you have something equivalent to a powerful staff member in a computer, or configuration of computers? This I think is a very cardinal idea. It was there from the beginning of the '60s when Licklider started the program, and he was interested in command/control and it was of interest to the directors of DARPA at that time. The seeds of the program were very much nurtured by this idea and then moved on. Now, interactive computing, networking, and the use of computers in time-sharing, emphasis on program development rather than on efficient running of a program on a machine, the time it takes and the ease with which a person can develop a program and change a program, flexible ways of handling that, these were more important in the DARPA environment than, say within the Department of Energy or in large national laboratories where the goals were to find machines of increased power in order to run faster certain kinds of complex physics computation or to have a much more accurate and proved valid compiler in order to take a large imperative program, say in FORTRAN, and run it. The attitudes were very different. The emphasis at DARPA was on how you can develop program development processes, flexible processes, by people together with machines, rather than on how you can run in a powerful way, with many cycles per unit of time, programs that are very well formulated and that change very little -- which is the most conventional way of doing computing. So, I would characterize the distinction between the DARPA style, which brought about many innovations in computing, and the more conventional style either in the large national labs or in business or in large companies, IBM for instance, as one of considering in one case the computer as an assistant with a certain kind, if you wish, of intelligence capabilities vs. considering a computer as, if you wish, a fairly low-level technician or servant or instrument or something along those lines. There is a distinction between the two. There are people who feel strongly philosophically about this conception - that it should be just a technician or a follower of instructions in strict detail. Should or shouldn't, the point of view at DARPA was not whether it should or shouldn't. The question was is it possible to develop that capability that could augment the facility that the person has in decision-making. It was an exploratory thing and AI was very much part of it. It was at the center of that, I would say. The other things were technical developments in computing that were essential for implementing the broader concept. For instance, DARPA was not interested for many years in the development of major computer power, having more powerful machines, or having more powerful compilers, or developing more programming languages. This was not very much something that DARPA was involved with. On the other hand, other agencies were.

With Strategic Computing, it is interesting that the whole thing was motivated by developing an AI technology that would be richer, and more mature, building on what was done over the last twenty, twenty-five years, that would have an impact on applications, in particular, military applications, that would be recognized by the services, that would also be used as an instrument for technology transfer from universities and research labs to industries, that would be used to help develop an industry of AI in the same way that an aeronautical industry was developed in this country. In addition to that, to see what are the other components of computing that had to be developed to make that possible in addition to ideas in AI. The technologies that were obvious that needed support were computer power to be able to support the AI ideas and algorithms and the appropriate software going with that, and also microelectronics, i.e., the VLSI stuff that goes beyond the technology of one, two, three microns that would provide the microstructures to support the architectures in the new computers. Initially, the motivation was that you needed the new architectures of parallel computing, because in supporting AI processes and algorithms you needed more power that only parallel computing can bring about. It's interesting that in the last four or five years, we found out that many of the central processes of problem-solving cannot benefit or exploit very well parallelism beyond something which is much less than a factor of one hundred. What we want is to be able to find parallelism at levels of 10 to the 6th or more than that, if possible, to try to get real power out of these new architectures. We found out, though, that there are other computations that are not to be found within AI, but they're conventional mathematical computations, PDEs and other algebraic computations, that can benefit enormously, from parallel computation, and that's what is happening so far. We found that certain deep mathematical computations can benefit fromparallelism and you can have weather prediction that you could not have before, or a modeling of material compositions, or simulation of air flow around an entire aircraft. All of these developments are important. But they were not initially the motivating force behind the parallel computers that DARPA had been developing. So now, parallel computing is going to be by itself a separate line of development within DARPA, independent of the need to implement powerful AI systems.

NORBERG: Does that suggest that the DARPA program has become more generally associated with computer science than it was in the earlier years?

AMAREL: Yes, I think it's becoming more generally associated with the discipline as a whole. There's going to be more connection with mathematics. I wanted to have more numerical computing within DARPA. DARPA did not have anything to do with numerical computing in IPTO in all the years up to now. I don't believe that numerical computing is not a part of computer science. There are lots of interesting things in numerical computing and optimization, but DARPA never touched that. On the other hand, there were other agencies that did support numerical computing, within computer science programs, like NSF, and OSR, and ONR, and Energy. So, you are right. I think I see DARPA moving more in the direction of more general support of computer science, including languages, programming languages. There was no interest in programming languages and more general software technology at DARPA up until very recently. But now I see DARPA getting more and more into a position where the STARS program is going to be part of DARPA, the SCI program is going to be part of DARPA; there is a major set of activities in software technology, including on the invention and formulation of high-level, wide-spectrum languages. It's growing. It's growing in directions that were not there before. It's nice ... it's interesting. But, you said... what's my view in terms of some of the initial couple of decades. I think that's what I believe was the driving force.

NORBERG: I want to ask two questions, but I'd like to lead into the second one by asking you to go back and elaborate on something you said in talking about the DARPA program. That is, what do you see as the innovations that the DARPA program has stimulated? Innovations in computing. If you had to make a list of them, what would it be?

AMAREL: I think, certainly, interactive computing, and the networking support for interactive computing. Much of the work in graphics in the country grew out of DARPA. The architectures of time-sharing and many things around those architectures, including some of the issues of protection, security and so on that go with time-sharing. Much of the technology of AI without DARPA wouldn't have happened. Even though much of knowledge-based systems grew up in the '70s under NIH support, this would not have happened if Dendral didn't start within DARPA and show that you can do something in a knowledge- base sort of way. Then the thing was amplified and leveraged through NIH, and then came back through Strategic Computing in other ways within DARPA and led to the development of an entire expert systems industry, which is a very important impact for the country. These things immediately come to mind as major innovations. Now recently, I would say parallel computing. Parallel computing is becoming a major activity at DARPA.

NORBERG: We don't know if it's going to be successful or not from what you said earlier about the problems .....

AMAREL: We don't know yet. We don't know yet. We don't know about the impact. There's no doubt it's going to be with us, but I don't know the impact yet.

NORBERG: Would you say that these innovations - well, this is leading the witness, but - Would you say that these innovations developed because of the determined focus of the DARPA program on augmentation of computing equipment for use by humans in decision-making? Or was the time ripe for things like time-sharing? After all, there were a number of projects in time-sharing going on before DARPA got into the funding of it. I can't quite say the same thing about networking. We can't be quite so certain about AI; it might have developed in the way it did, maybe more slowly to be sure, but...

TAPE 2/SIDE 2

NORBERG: Do these innovations depend upon each other in such a way that they are self-reinforcing? If you get one you're going to get the next one relatively quickly because of the sharp focus of the program?

AMAREL: I think the fact that there was a vision and there was an orientation were important. I don't know if they were determinant, but they were very important. Among other things, they concentrated the resources in a certain area over a relatively long period of time. They chose among the different things that could have been done some few things that were consonant with, were related to, the overall vision. Graphics is important for interaction, for easy, graceful interaction, between man and machine. Networking is essential for access. Time-sharing is essential for utilizing rare resources through networks - people who don't have them locally would be able to use then. All of those things are parts of the same overall conception of a way of doing things. Doing knowledge-based problem-solving is part of the concern with developing an assistant that will do decision-making for you. Yes, I think that a conception, a view, an orientation, a set of goals is important. I think an agency like NSF does not have that as a matter of charter; it doesn't guide the field; but it is primarily responsive to ideas that emerge in the scientific community. To summarize, I think that the DARPA guiding principles, the continuity of support, and the concentration of energy have contributed enormously to the major developments that were promoted by IPTO. Yes.

NORBERG: Then do you think that in the past four or five years the IPTO program, or ISTO program, as it's now called, has that same sort of vision?

AMAREL: Certainly during the time I was there, I was very much aware of this vision, very much took that as a part of my five-year plan, my goals and so on. These were the over-arching goals. I'm not quite sure what happened after I left in '87, whether there was as clear a view, whether these remained the guiding goals. I think it's best if you discuss that with Jack Schwartz. But my sense was from interactions that I had with Jack that he felt that you have to be more sensitive to the state of various sub-fields in the science and try to, as much as you can, exploit sub-fields that seem to be ready and ripe to be pushed further to a point where you can have a lot of impact. So I found from the many discussions I had with Jack that he gave more emphasis to the state of a sub-field or a field and he concentrated on those sub-fields where the added value that he could provide, through some additional support, would make a significant difference. This is a rather different attitude from the following: here are goals you want to reach, let's try to put a lot of energy to try to reach those goals. It's a different view of the world. He and I have had some discussions about all this. His metaphor is "waiting for the wave." You wait from the side for the wave to come and when it comes you ride the wave, rather than swimming very hard against the current to try to get someplace because you think it's very important to go there. I think DARPA does both things; but sometimes swimming very hard against the current to get to an objective is very important. It is also very much the dominant DARPA approach. Sometimes it brings successes, sometimes not, because the discipline may not be ready yet. You may have some very basic hard problems that you're going to find in your way and then you may have to say, okay, that's it, it's not the right time. But how long does it take to know whether you have a hard problem? This is again a major issue of judgement. Is it one year or three or five? Well, in many of the problems that DARPA explores, you cannot do things in one year. You have to allow three years to see what the difficulties really are. But then you have to do something about it; you have to get the PIs together around a table and find out what is the state after three years, whether it looks like we are moving or stagnating. If you are stagnating, you just do something else. DARPA is completely equipped to do just that, to drop something after three years or five years and to do something else. But then you have from time to time to have these risky investments that seem to be going in the right direction. If you believe it's going in the right direction, there will be enough people who believe so too, and there will be a few scientific developments that can give you a base on which to build; then you go and do it. But don't cut it after three months or one year; let it go for a certain amount of time, then get people together to see where you are. I've seen this happen during the time I was there and I've seen that happen before.

NORBERG: How about some examples of that? What sorts of projects did DARPA fund that you would say were swimming upstream?

AMAREL: Speech in the '70s. They wanted very much to try to achieve major progress in the technology of speech understanding. It was hard; it was a very hard problem. After five years of intense work they decided to get completely out of it. Actually, I disagree with the way this was handled, because the result of those five years of experiments, at least in a couple of the approaches, were very promising. The work could have been given a few more years to mature further. But at that time the effort was dropped; now, work on speech started again in the '80s. Now, it's moving again. It's moving okay; it's moving well.

Another thing that was swimming against the current was parallel computing in the '70s, the work at Illinois. There was no preparation yet with what to do with this kind of computing. There were microcircuits; there was the technology. So this was interesting work showing that the technology could be used within the new architecture.

But the architecture by itself did not find a niche anyplace. It could not be used. It's interesting that at that time it could not be used profitably even in the solution of partial differential equations; I don't understand why not. It's an interesting issue to know why at that time the people with the large problems did not jump for it. NASA did to some extent; right, you focused on their work. NASA did use actually that work in the '70s to some extent. There was not the same kind of excitement that I see today with parallel computing by the numerical computing community. In the '70s the situation was much, much more difficult. So the parallel computing effort of the '70s was dropped after "swimming upstream" for several years.

The entire area of robotics within DARPA is a fascinating one. There were many ups and downs. There were times where DARPA pushed in the direction of trying to develop special manipulators, walking machines, special kinds of robotic devices. But there was never a consistent program in this general area. It went up and down. This needs a special study. I don't quite understand why this effort never had a steady orientation, that it was so much fluctuating -- with a period of the order of three years, roughly. During the time I was there, there was a fairly reasonable, good program in the area of robotics, but not very large, on the order of eight to ten million dollars. In addition, there was a program on the autonomous land vehicle, part of Strategic Computing, which was also a promising program, but, again, it didn't have enough continuity. A year or two after I left, the program was abandoned, after it was found that the army didn't show obvious interest in the program in the long run. That program, the entire robotics program, has not been consistent and steady. I can't understand that. This needs a little more thinking, a little more studying. Are you going to talk to anybody in the robotics area?

NORBERG: I don't know yet.

AMAREL: I think Reddy would be a very good person.

NORBERG: Okay. I'm leading toward two things here at the same time with this discussion. Would you say that the examples you have just given about speech and ... well, speech doesn't fit it, but the examples that you've given are outside the augmentation idea, that is developing systems for assisting humans?

AMAREL: Speech was in the direction of augmentation.

NORBERG: Speech was, but parallel processing, modules, robotics and so on. Do those things fit that?

AMAREL: Not really for assisting decision-making. Robotics was not.

NORBERG: Now I want to do the coup de grace here. In the case of speech, you said the field wasn't ready for that yet. Is it also the case in parallel processing and robotics that the field wasn't ready to accomplish the goals set for those programs at the time those programs were very active?

AMAREL: Robotics is a separate issue. But the issue of parallel processing was premature, especially in the applications part, not as much the VLSI part and the small structures part. The applications part was not there. There were no people in applications who (1) wanted to have this kind of power, who (2) knew what to do with this parallel computing stuff. There were enough alternative ways of doing complex computations that were superior in terms of ease of using them and so on, to justify the effort of finding good ways of using an array of 64 processors. It was not a major change in the amount of computer power that you could get with conventional architectures. You could've gotten it with a sequential computer at less cost and at less cost in developing the software and in the development of computational paradigms. But, if now, you have available 64,000 computers working on a problem or a million computers or even 500, it's a substantially different kind of computational power relative to sequential architectures. Under these conditions, it makes sense to invest in developing software and appropriate computational paradigms for making the new architectures usable for your problems. And maybe the problems that you are prepared to solve require that kind of power. The problems that people were prepared to approach in the '70s did not require that kind of power. There were no people screaming for this kind of power at the time. Now, if you look at the grand challenges that came up in the OSTP proposal, you'll find that several of those grand challenges will require this kind of computation, parallel computation. Have you seen the list of grand challenges of OSTP?

NORBERG: No.

AMAREL: For instance, let me give you an idea: prediction of weather; material science; semiconductor design; superconductivity; structural biology; design of drugs; the human genome project; quantum chromodynamics; astronomy; transportation problems (which means mostly design of particular aircraft, even cars and ships, in which I told you I'm interested); vehicle signatures (for example, in the case of submarines); vehicle dynamics; nuclear fusion (fusion research always wanted to have lots of computing); efficiency of combustion systems; enhanced oil and gas recovery; computational cognitive sciences; speech; vision; undersea surveillance for ASW. This is a set of grand challenges that was compiled by OSTP as part of the most recent version of the federal high performance computer program in which I was involved back in '87. People are thinking about these kinds of computational problems now. They really want to do something about them, and we didn't have this attitude, I think, 15 years ago.

NORBERG: One of the reasons often given for why ILLIAC IV was not seen as a significant machine was that people had to do their own programming for it, whereas the principal competing machine, the CDC 6600, you could use languages like FORTRAN and so it was much easier to use it.

AMAREL: We have the same problem now.

NORBERG: Now that would be an explanation of why people weren't screaming for the power. Is that fair to say?

AMAREL: But we have the same problem now. People still don't know how to program the parallel computer to solve their problems. There is much more progress in the hardware architectural part of parallel computing today than there are advances in the software needed to control the operation of these computers so that: (1) you can minimize the overhead of communications and (2) that you can have as full utilization of the available processors at any point in time. We still don't know how to do this kind of stuff well. We don't have good languages for doing these things. There are experimental languages. You have some parallel FORTRAN that is being developed, but this is still at the research stage. I am not quite clear if I could characterize the differences between ILLIAC and now. It's not clear. There are all these interesting issues that come into that. There is the existence of demanding applications -- of problems, and the pressure of people who want to solve them now, who believe that the time has come. There is the question of software availability and whether people know how to utilize the parallel beast now vs. how much they knew how to do that then. The cost must come into that. At this stage the cost of microcircuitry is different from the time of ILLIAC, and to get 500 processors today is a reasonable thing to do. At that time, it was not a reasonable thing to do, while at the same time you want everything to be reliable. Now the technology is more reliable. There are so many things that come into that, but there is no single obvious thing that I can say about why now this seems a better time to push parallel computing vs. the '70s with ILLIAC. I'm not sure about the history of ILLIAC. It would be interesting to look at the details of the history and what were the triggers that brought about that project. I think the director of DARPA at the time had much to do with that, Lukasik, who was very much taken by this project. It would be nice if you could talk to Lukasik because he is still very much interested in parallel computing. I've had conversations with him on that. He doesn't think that it was a bad choice. He doesn't think it was a bad decision at the time to get into the ILLIAC project.

NORBERG: Well, it may not have been a bad decision, but it had a bad outcome.

AMAREL: All right. Yes.

NORBERG: I'd go that far with it, myself.

AMAREL: Did you follow it, the ILLIAC IV?

NORBERG: Well, I read the history of ILLIAC IV, which was published by what's his name from Illinois, which is a bit of a problem. There are several assertions made about the influence of it that I need to check, because it just doesn't sound right to me. He talks about the logic circuitry as being influential. He went as far as to say the CRAY 1 was influential, and I mentioned that to some of the people at Cray, and they didn't even know what I was talking about, let alone whether it was influential or not, people who work on design.

AMAREL: You mean the work on logic circuitry in the ILLIAC was influential?

NORBERG: That's a claim that's made. In fact, that is also mentioned in Bell and Newell. So, they picked it up from the history of the ILLIAC.

AMAREL: Could be.

NORBERG: It could be, but we need some confirmation, that's all. I've got to find some people in Burroughs to tell me how it affected their work. I'd like to continue that, but I want to get at it from another point of view for a moment. Several times in the course of our conversation today, you've talked about command and control as being a very important element in guiding some of the activities that DARPA was involved in. Does that suggest that the program in DARPA had a military-oriented cast to it all along? That is, it really does have a mission objective; it's not basic research as we know it in NSF and NIH and places like that.

AMAREL: There's no doubt there was a mission objective and that this particular goal of facilitating and strengthening command/control was very definite and it was very explicit. It appeared in documents, I'm quite sure, certainly appeared at the time I was around. There is always a distance between asserting this kind of top-level goal and making choices of specific programs, of specific scientific orientations. Now, you have to make an argument why this or that technology, be it networking or certain parts of artificial intelligence, are appropriate in light of the top-level goal. As a matter of fact, frankly, I believe that almost anything that has to do with computer science could be shown to be appropriate to this top level goal. So there's a strong element of judgment about what things are more or less appropriate relative to the goal. Command/control requires computers, requires communications, requires quick digestion of lots of information, summarization, storage, analysis, simulation capabilities. You'd like to be able to come up with support for decision-making. Almost everything in computing can be a requirement for command and control. In my view it's a very weak kind of pulling goal. Very weak. I mean weak in terms of permitting selectivity and differentiation at the program level. It depends on how you interpret it and where you put the emphasis. I think what happened in early times of DARPA is that the emphasis was put on some of the higher levels of command and control tasks, that had to do with the information processes that go on in decision-making and in problem-solving, rather than at the level of circuits and electrons. Slowly the emphasis has been moving in the direction of the systems that support the high level processes. Often you want to do experiments, to show feasibility of an idea, so you must show that you have a computer that implements the idea, a computer that has an architecture which can support the processes that you are studying. Because of this, you must develop the computer and the network and the software to implement the higher level processes. You need to experiment with them. It is not enough to wave your hands. Many of the systems oriented projects, I think, developed as part of the program because they were needed to provide experimental environments, to show feasibility, or to show how the program should be varied because of possible difficulties with an idea. But they were not driven primarily by technology. They were driven by the need to explore a high level process. It may be that later on the technology became more of a driving force. It happened in each of the sub-areas in the office, in networking, time-sharing, and in parts of AI.

NORBERG: If we look at the early years and Licklider's coming to IPTO, Licklider's vision is larger, more compatible with a command and control problem set than the people at MIT, for example, who were just interested in time-sharing and improving the capability of programmers and people who were interested in using machines on an interactive basis, particularly in AI. That Licklider's view of the world was much more compatible to the military issues in command and control, and therefore that drives the program from 1962 on in DARPA and continues to give it some sort of spirit, until maybe Strategic Computing in 1982.

AMAREL: Strategic computing is still compatible with this entire idea. Very much so, because it has to do with taking out of the lab AI and trying to make it work in realistic situations. It's compatible, except that now it has been changing in the last couple of years, because computer power, very large capacity computing is becoming by itself a driving objective, which was not there at DARPA up to almost now.

NORBERG: Let me address that question from...

AMAREL: I wanted to give you something....

NORBERG: This is called the ISTO Strategic Plan dated 24 June 1987.

AMAREL: And another paper that has to do with my own view of themes and issues in information science and technology, that I put together in February '87 during the time that I was there. I thought this might be interesting to you.

NORBERG: Yes. Now, I was just going to turn to that and your time at ISTO. How were you recruited? What was your first interaction with DARPA, the IPTO program? I'm thinking really of funding and involvement.

AMAREL: Funding from DARPA goes back to the mid-seventies. I always had connections with DARPA, but not in the relationship of somebody being funded. Mostly because many of my friends in AI were supported by DARPA. But in the early '70s, I did go and try to get some support from DARPA for doing work in AI. At that time, one of the program managers was Cordell Green, who was there for three years or so and then he went back to Stanford; now he has his own company called Kestrel, in the Bay area, doing R&D in software technology. Well, I knew Cordell when he was a student at Stanford University, and he heard my lectures and so on, and he said I would like to find a way of funding you, but at that time there was no way of getting additional funding in the AI area as such. So we did get some funding in two other areas: in automatic programming and in database security. I was involved as PI, and some of my colleagues here at Rutgers did database security. I was not directly involved in that, but I was very much involved in the automatic programming part. It was very close to AI. It was very close to my notion of starting with a very weak program that could be based on schemata of problem-solving that come from AI, and gradually transforming that program by introducing specific knowledge from the domain to make it a strong program for solving problems in that domain. So I was involved from '73 perhaps to roughly '78 in DARPA-supported activity. In '81 I went again to get some support. The program manager involved at the time was Bob Englemore. I was very much interested in work in the area of AI and design, AI and engineering. As a result of these interactions, we did get some support for AI and VLSI design. I was involved in the first year or two, and then the program went on with a colleague of mine, Tom Mitchell, then Lou Steinberg, as PIs. I had another involvement in the mid seventies, which is part of a policy involvement. For four years in the '70s, I was member of a group - the Jasons - that worked in the summers at La Jolla on various studies that are significant to national security. One summer, '74 I think, or '75, they were commissioned by the director of DARPA to look into the AI program and to come up with recommendations about the scientific soundness and significance and the promise of the program.

NORBERG: Was this Heilmeier?

AMAREL: Heilmeier. I was a member of a group of a few people from the Jasons, less than four or five, who interviewed people in the discipline and tried to put together a report in this particular area, with a full understanding by my colleagues over there that I was very much a part of the AI enterprise myself. It was a very difficult time, because AI was on the chopping block at that time. As a result of the summer study nothing concrete and specific came up. We did not go back to the director of DARPA suggesting that something specific, A or B, has to be done; except we indicated that AI is a promising discipline that has achieved some interesting things so far, and we are not in a position to say much more about the promise for the future. This was roughly a two-page report. It was a very brief report. So I was involved then in a policy mode for a summer study of DARPA programs. At that time I was quite close to what was going on inside DARPA. I knew quite a bit about the programs of AI in the '70s. This was a crisis time at DARPA in terms of AI and where it was going.

NORBERG: Why?

AMAREL: Because at that time the DARPA director was questioning the soundness of the program and whether the program should exist at all, or what parts of the program should exist at all. And, as you know, after he left as director, he became extremely supportive of AI as a discipline and of specific work in the field, and by now is vice-president for engineering in TI, and has been very strongly supportive of AI. But during that time in the mid '70s he was quite a bit of a skeptic, and he actually felt that maybe it was a waste of resources for the agency to continue doing something in AI. The interactions we had at Jason were not on programmatic grounds, but rather on what was the science of AI, what is going on in the field, etc. So, it was very interesting by way of eliciting a variety of interesting views. Newell was one of the people who came to the summer session and Minsky and McCarthy and Moses came, and Winston and Lederberg and Feigenbaum.

NORBERG: Were they all members of the Jason group or just assembled for this AI study?

AMAREL: No, no. People came as invitees of the Jason group to make presentations and to interact with members of the group. But we at Jason put together the final report. So these are the ways in which I was involved with DARPA before '84-'85.

TAPE 3/SIDE 1

NORBERG: So, you've come up to '84 now. How were you approached by DARPA to become director of ISTO?

AMAREL: I was on sabbatical in early '85, at Carnegie-Mellon again, working with Simon on special problems of AI and scientific discovery. Up to that time, I was chair of the department here at Rutgers, and I decided at that time that this is it; fifteen years of being a chair is enough. I was sort of in a transition stage. That was essentially one of the reasons that I took the sabbatical. It happened to be the time at which the director of DARPA, Bob Cooper at the time, was looking for a replacement for the IPTO director, Bob Kahn, who was director for seven years and who also felt that he was long enough director of the office and wanted to try other things, wanted to get out and do something similar to DARPA in the civilian world. So there was a search for a new person to direct IPTO. Cooper invited a group of about fifteen senior computer science people in the country to advise and help him in the search. Most of these people were PIs of projects that DARPA supported. Most of them represented centers like MIT, Carnegie-Mellon, Stanford, Columbia was there, Berkeley was there, several of the other laboratories, Lincoln Labs, ISI, and so on.

NORBERG: Were you present?

AMAREL: No, I was not in this group. I think actually all the people in that group were heavily supported by DARPA; they represented the major laboratories that were supported by DARPA. I heard from Bob Cooper himself when I was in Carnegie-Mellon and from some of the people who were members of this recruiting group and whom I knew very well; they tried to convince me to come to DARPA for some time as director. So I was actually recruited on the basis of a recommendation by a select group of colleagues in the country. This kind of thing happened before when there was a question of deciding on a selection for a DARPA recruit; the director usually consulted with a small number of people in the community before making a decision. In my case, there was a broad consultation. That's the way I was brought in. This was not the way in which my successor was brought in. There was no major group of people who advised the then director of DARPA. But in my case it was done that way. It took some time for me to decide; I was not completely sure that I should accept the offer, and there were some problems in Strategic Computing that I was concerned about, but I decided I'd give it a try. That's the way it went.

NORBERG: What was the program like when you arrived?

AMAREL: There was one major problem at the time, which I was very concerned about. In organizing the Strategic Computing program, at some point the DARPA director decided to separate the applications part of Strategic Computing from the technology base part.

NORBERG: Can you describe the difference for me please?

AMAREL: The problem was roughly a hundred million dollars a year. At that time it was a bit less. Now it's around a hundred and ten, a hundred and fifteen million dollars a year. But, roughly, 75%, 70-75% of the program was directed to the technology base - technology base meaning research that was generic and not directly oriented to a particular application. For instance, in the case of speech, general research on issues of phonology, and in the case of vision, research on visual features such as texture, on model-guided matching of figures, on understanding vision in motion - basic scientific technical work in areas of interest within the subdisciplines of Strategic Computing.

Another example is software environments for development of knowledge-based systems in AI. Not necessarily a particular knowledge-based system for a specific application, but more general, generic sort of systems. The idea of generality and generic systems was very much behind the concept of technology base. In the area of computers and architectures, the idea was to explore what you can do with message passing in parallel computing vs. shared memory. What are the problems in an architecture such as connection machines? Can we move connection machines from 64k to a 1,000,000 systems? What are the problems? What are the problems of programming those beasts? In VLSI there were similar problems as to how to move to 0.3 microns of technology, how to do gallium arsenide, how to do opto-electronics. It went all the way from the physics of devices up to the architectures and up to problems in AI and speech, vision, and the building of knowledge-based systems - roughly around 60-70 million dollars of Strategic Computing program funds. The remaining funding went into several applications: the autonomous land vehicle, the pilot's associate, the battle management system for the Navy, and more recently a few other projects with fairly well defined goals. Initially, Cooper decided that the way to organize this enterprise was to separate the technology base from the applications. And also to separate organizations. So another office was created, which was called EAO, Engineering Applications Office to handle the applications part of the program. IPTO was the office that I went in as director. It was to handle the technology base of the program. As the time went by, it became clear to me that what we were going to have is two information processing departments because the applications office would have to develop its own technology-base to be able to support some of the things they were doing, and many of the people in the technology-base activity would have to be much closer to some reasonable applications to try their ideas on realistic testbeds. The organizational differences created some territorial frictions. From the beginning, my recommendation was to fuse those two offices, to bring them together.

NORBERG: Were they separate when you arrived?

AMAREL: They were separate. I was director of IPTO. Clint Kelly was director of EAO, and the deputy of Clint Kelly was Craig Fields, who is now director of DARPA. Craig has been director of DARPA since April. So, Clint was director of EAO, with Craig as deputy, and I came in after Kahn as director of IPTO and I had as my deputy Bob Keagans, an Air Force officer who came around the same time as me at DARPA. The group at IPTO was covering everything in information processing, from networking to VLSI to a certain amount of robotics and then quite a bit of AI and also the entire area of computer architectures and the software program. One of the reasons I went to DARPA was because I felt the Strategic Computing program was a unique opportunity, a tremendously important opportunity to do something important in AI. It had to be done carefully and well. It had to be taken seriously. I didn't see that as a military program, I saw that as a program that was feasible within the context of defense, and I thought that it was the right way to do it. If 70% of the program was science and technology, that was great. The science and technology was to be good quality, and it had to be advanced. It had to be well connected to its applications, no doubt, and the organizational distinction, i.e., the separation into the two offices, made it difficult. So I campaigned from the beginning for a fusion. Eventually, in May of '86, the new director of DARPA, Clif Duncan, who came after Cooper, decided -- after several false starts -- to fuse the offices, to have IPTO and EAO in a single office, to name the new combined office ISTO, to have me as director of the fused two offices, to have Clint Kelly as executive director in my office, to have Craig Fields go as Chief Scientist to the front office with him, and to see how it goes. The creation of ISTO via fusion of IPTO and EAO was useful, but it was not completely resolving the problems of Strategic Computing, because other offices at DARPA had pieces of Strategic Computing. Another office was responsible for the pilot associate application. Still another office, the Navy office, was in charge of the battle management one. There was a piece of mathematics that was supported by Strategic Computing that was in the Defense Sciences Office, still another office. Again, it was very difficult to have a single person to be really in charge of the Strategic Computing program as a whole, with both budget responsibilities and technical responsibilities. So I found that difficult throughout my stay at DARPA. It was one of the difficult parts.

NORBERG: What sort of problems were created as a result of that?

AMAREL: The quality of some of the applications programs was not as good as it could be. The input that the applications programs got from the best people in the field (who were mainly working on technology base projects) as to how to develop knowledge-base systems was very limited because the application contractors worked independently with their own engineers and the people in the discipline that would best permit them to keep up schedules and deadlines. The issue of milestones and deadlines was very important to the program. In general, I felt the quality of the program was not as good as it could be. I saw two main reasons for this. One was that the applications themselves could not benefit sufficiently from developments in the technology base, so they could not have the kind of quality that would reflect the present best state of science and technology accurately. The other reason was that the people in the technology base did not have the benefit of working in close contact with real-life problems and of interacting directly with the experts in significant problem domains. They had to go through "interpreters," through several layers. I would have liked to have looked at the entire program and organize it in a different way. This was one of the important reasons that I found it challenging to go there. The creation of the fused office was a step in the right direction. In time some of the programs, from my point of view, were doing quite well.

I was concerned with funding problems throughout the two-year period that I was there. There were constantly reductions in funding levels, from what was projected or expected or planned. In many cases, computing was the place where one could find quickly certain unexpended funds that would make the base for new agency programs to be started quickly. For instance, that was true in the case of the national aerospace plane that started during the time I was there. It was true in a couple of other programs, one of which was armor and anti-armor that the deputy director of DARPA, Jim Tegnelia, was very much interested in. Money for these new programs came from various DARPA offices, but ISTO was a major donor. So one of my problems was the constant need to defend budgets, to try to keep them at reasonable levels. Sometimes I succeeded, but not always.

I wanted very much to have some new initiatives start during my stay in Washington. In the middle of '86 I started thinking of a general initiative in the area of computer-aided productivity (CAP) that would have areas of design and manufacturing as targets in the same way that applications of AI in the military were targets in the Strategic Computing program. A CAP initiative would be of similar kind to Strategic Computing, at the same level -- roughly a hundred million dollars a year. It would provide another focus for activity in computer science, where AI would be an important component but not the only one. Simulation, very high capacity computing and databases would be other major components of the effort. Unfortunately, the CAP initiative didn't go beyond the level of initial exploratory plans, and some analysis of the plans. Actually, I left in September of '87. In August '87, we had a major workshop in

Monterey on the West Coast, in which several days were taken by an analysis of this proposed program. Lots of people came up with ideas about how to actually implement it. But, unfortunately, I was not in Washington to pursue the proposed initiative into the implementation stage. I would have liked very much to see this program materialize.

Research on networking continued very much at the same levels as before during the time I was there. It was the beginning of the development of ideas about the national research network that now appears as a key component in OSTP national plans for computing. I was very much involved in the development of the OSTP strategy report on very high performance computing. I was the chair of the R & D committee of that particular activity in Washington, and networking was a very important part of that.

I think that the major new developments during the time I was in ISTO were in the systems area and in the software area. In systems, there were some interesting developments in parallel computation. In software also there were some interesting developments; in particular, there was the beginning of an independent software program within DARPA, which was not there before. I brought in from Carnegie-Mellon an associate professor, Bill Scherlis, who came from his university on loan for two years (same arrangement as mine), and now he's extended that for another two years, to manage the software programs. He's a very good man, and he had many good ideas in this area. He wanted to push in the direction of automating, as much as possible, the program design process, which is a very difficult thing. He has been going bottom up from engineering and infrastructure and common utilities and tools in the direction of conceptual problems. So, that was a good development.

NORBERG: Now, this is outside of AI, isn't it?

AMAREL: All of those things are outside of AI, but AI is only around 20-25% of what goes on within that office. I think it's an important part, but as director, I was interested in other areas as well as in AI. The budget of AI during the time I was there in '87 was of the order of 45 million dollars a year, a little more than 25% of the total office budget. In addition, there were applications as part of strategic computing where AI was an important component. So if you add applications, the actual support of AI was higher. I could give you later on some more numbers about budgets in AI from the period from '83 to '87-88.

NORBERG: What I was driving at was getting back to the other issue that we left aside for a moment a little while ago. That is, it sounds to me like during your administration there, this shift away from augmentation into other areas of support for computer science was already going on.

AMAREL: Yes. But, you see, I consider software and software technology as a very integral part of that. I mean, the development and use of very simple and powerful environments for program development is part of that. We found that in the development of knowledge-based systems in AI what is needed from the systems point of view is good tools for putting together, very quickly, knowledge bases and inference engines. Software technology is a very important part of that. It's as important for developing AI systems, as it is for developing operating systems for time-sharing or for developing networking protocols or for numerical computing. There was no numerical computing in the office during the time I was there. There's still not now. There is nothing along these lines. But software technology, I believe, was a very important thing, a high priority in my plans. It reached a level of about 25 million dollars a year when I was there. It is bigger now. This is an area that both Jack Schwartz and I pushed very consistently and I believe it's a very important technology to develop. It is going to become now more and more a central part of what goes on in that office with Barry Boehm coming, whose main interest has been in software over the years. The business of parallel computing already started before I arrived. This was something that started in the beginning of Strategic Computing. I thought that it should continue. My only prejudice was that there was no balance between the hardware architecture development and the software and algorithmic kind of developments. If at all, my only correction there was to try to increase the amount of software activities in the area of parallel computing and to encourage algorithm design and theoretical work in the context of parallel computing. I didn't believe there was a good enough balance between the tool and ways of using it. The balance is still is not there, and it is still very much needed. I was not looking at parallel computing as a nice, elegant piece of technology by itself. I wanted to know whether we could *use* it in some way, in interesting places. I still don't find evidence that we know how to use it very well. It's not only because of some kind of limitation in ideas, and so on. I think it may be very much intrinsic to the nature of computations that we try to implement in parallel. Computations in symbolic computing and in AI problem-solving are very data dependent in the sense that you have to wait for results of computation before deciding what to do next. You cannot do them in parallel because you have to wait until an input arrives from another part of the system. You have to be much more context dependent; the context is important. You have to constantly find out what the other parts of the system are doing. These kinds of gathering together, from other places into one, is always a bottleneck in cases of parallel computation. I think, fundamentally, we are going to have limitations as to what we can do in parallel. We'll get to the point perhaps in a few years, where we're going to know what kinds of problems are amenable to parallel machinery and what are not, and then we will have good ways of allocating sub-problems to appropriate kinds of architectures. I think by the mid-nineties, by the end of the century, I'm sure we're going to be there.

NORBERG: Saul, do I understand correctly that you were unable to institute any new programs while you were there during that two years?

AMAREL: Within AI, I think programs in the area of planning and scheduling were new; I brought them in. Within software technology and within systems, things that have to do with software design and with algorithms for parallel computing, I brought in. This is pretty much it.

NORBERG: Considering those examples, can you take me through a budget cycle at the time you were head of the ISTO office, from the conception of a plan through to granting money to somebody? How did that go? From whom did you need authority? What was the process of getting proposals? What was the evaluation? What were the criteria you used?

AMAREL: The plans grew up internally because every program manager is sort of an investment banker with a portfolio, with laboratories he works with, with a certain orientation about technology and science. So there were inputs from each one of the program managers about the substance of his program as he wants to see it next year and his requests for budget levels: one for continuation of existing programs, two for new initiatives, fairly detailed. This would go on around the end of the calendar year, the beginning of the next calendar year, and would be the basis for a draft budget for the office. I would get some indications from the front office about the rough order of the bottom line that the office can expect. Nobody took that completely seriously, but seriously enough. So everybody took that as a lower bound, usually more than that in budget presentations. A proposed budget was presented on a one-day basis, full day, to the director of the agency. There were three layers of authority. The program manager, the office manager, the director of the agency; so there were three decision making layers. The director of the agency would sit for an entire day, listening to budget presentations, which were not just fiscal. You have lots of meat in these budget presentations. There were discussions back and forth. There would not be any decisions on the spot. But this would be a review process; then the director would maybe invite you for some clarifications, give you a sense of the things that probably are going to be killed, things he'll leave in, and then you get an edict from the director that this is your budget for the next year. That's not the entire story. There are many, many other kinds of cycles that went on at the top level because of changes in budget appropriated by Congress or because of various kinds of budget reductions at the DOD level - 5%, 7%, and so on. So what you were usually requested to do was to order the various elements of your budget. For instance, a particular program manager, let's say in AI, would have 45 million dollars. This would be broken down into pieces, not into projects like the research at SRI or at Stanford and so on, but into elements such as vision and within vision there would be three sub-categories; speech, with two sub-categories; knowledge-based systems, with several sub-categories, planning, and so on. So, he'd have something like ten sub-categories and there would be a budget allocation to each of them and also a priority ordering among the sub-categories. Then a total priority order of all the program sub-categories within the office would be developed, where you would have a bunch of pieces in AI here, a bunch of pieces in networking up there, a bunch in infrastructure, and so on. This was the biggest internal exercise, this prioritization. When you had to handle a budget reduction of say 5%, then you cut off the bottom 5% of the prioritization list. Eventually the program managers were told what is the situation in terms of their budgets, and they had some latitude within those budgets to change certain things. Of course, every particular action (i.e., an order to contract research funds for someone) would be signed by a program manager, would be signed by the office director, and then it would go to the front office and require the DARPA director's signature also. You had to show, in each case, that it was within the agreed-upon budgets and priorities. The actions between a program manager and a particular research project, let's say with

Newell at CMU, are of a different kind, and they have their own timing. It depends on the timing of a particular research proposal -- for a new project or for renewal.

In the distant past, proposals were solicited from particular institutions or researchers. At present there are no sole source proposals at DARPA. The whole thing is done with broad area announcements. When you put out a broad area announcement that you want proposals in certain areas of artificial intelligence, say, you get n proposals and then you organize a selection process for choosing meritorious proposals. Let's suppose that Newell's proposal is among those selected, then he's asked to interact with a program manager on budgets and on a statement of work. Never was there an unmodified proposal that I've seen. There are always modulations of a proposed project in terms of the bottom line, the timing for different pieces of the budget and items in the statement of work. You've probably had the same experience yourself with your own proposals. This process takes place between the program manager and the PI. When there is convergence on the proposal, the program manager had to find an agent outside DARPA who would initiate and monitor the research contract. The agent, who would be from an agency such as ONR, AFOSR, etc., would be involved in the administration of the contracts, even though the program manager would be always there as the key person who makes judgments about the scientific quality and programmatic quality of work on the project. He would go to site visits and so on. With a time delay between six months and nine months from the time a proposal is submitted to DARPA, the funding would be instituted with the researcher. This is roughly the situation.

NORBERG: It sounds to me, the way you've described the budget possibilities and the prioritization and the structuring of the budget and so on, that there was little room for new ideas to come into the office.

AMAREL: 10%! Maximum latitude that I had for introducing new program elements and changes, the maximum that any director has in the agency, assuming a fairly steady state situation. My problem was that I had less than 10% because most of the times I had to handle reductions in the budget. The budget was almost completely dominated by commitments from previous years. Unless you were prepared to have some radical kind of excisions...

TAPE 3/SIDE 2

NORBERG: So, you didn't have any latitude then, for doing anything, outside of this 10%, unless you could reduce some contract somewhere else - that's where the tape let go there.

AMAREL: Oh, yes. That's true. The effective latitude was small. When the budget is reduced, then you have to work with a budget that just covers continuation of previous commitments. Then, the only thing that you can do, in order to introduce any innovation, any change, is to reduce some of the previous commitments or stop them completely. This is always a very hard thing to do. For me, it was extremely hard. I found that there are some other office directors who didn't have too much difficulty with that. I also found that this is exactly the area in which most of the politics with the front office come in. You find that many presidents of universities and senators and congressmen are calling the DARPA director exactly when there is this kind of event happening or about to happen. Most of the interference, the micro-management from the front office, came up in situations where it became impossible to go ahead with the proposed reduction of certain kinds of projects, because of political pressure considerations. Under these circumstances, the office director or the program manager could not do very much, and the small latitude for innovation that was available became further reduced.

NORBERG: Did you ever try to reduce anyone's grant/contract?

AMAREL: Yes.

NORBERG: Did you succeed?

AMAREL: Yes. In some cases I did. Especially when there was no major interference from the front office. In cases like that, it was always clear that the project to be reduced or terminated did not have as much value as some new project that was considered for initiation. But there was always an enormous amount of resistance to any kind of reduction of a continuation. It's much easier to continue something in an environment such as DARPA than to start something new, much easier.

NORBERG: Can you give a percentage, a distribution of how you spent your time when you were at ISTO? How much was spent out in the field? How much was spent in the office? Of the time in the office, how much was really on the work of the office, and how much was dealing with these political problems?

AMAREL: Roughly 20% of the time out in the field, I would say. Two days a week on the average.

NORBERG: That would be 40%.

AMAREL: No....yes, no. I would say one day a week. A program manager spends more like 30% in the field. There are visits in laboratories or conferences with PIs and so on. I would say one day a week. Then, for planning and policy issues, let's say planning and internal policy issues, another day; coordination with other agencies and other discussions within the government, not only within defense, another half a day; local territorial wars and politics, another day; and the remaining part was really honest-to-goodness scientific evaluations and reading of proposals and discussions and interviews with people who had ideas. Lots of people who had ideas came in day after day. I found it very important to be able to interact with these people, without necessarily having any commitment that this idea is going to lead to this or the other funded project. That's one of the important things that an office director has as a plus, which is, the perspective that comes from lots of people coming and telling you about their ideas, not necessarily at a final stage of proposal development, but rather at an early stage. I would say it was of the order of one day a week, which was mostly in the office that I had interactions of that kind. Another half a day of discussions with program managers on scientific matters, merits of proposals, and looking and signing documents, roughly this kind of interaction. So, you do have one day of field work, visits and so on; you have another, similar day with people at the idea stage in the office, another half day with program managers, and the rest is a variety of political and coordination activities and relating-with-other-agencies kind of functions.

NORBERG: Did you rely on the program managers to make decisions about the quality of proposals that were coming in, or did you help to make those decisions?

AMAREL: To a great extent, I did rely on the program managers. In some cases, I went very carefully into a proposal when I found there was a certain amount of hesitation. I didn't read carefully all of the proposals. I read most of them, certainly in areas in which I was quite familiar, not only in AI, but in other areas - in software, theoretical computer science, some work in algorithms, not as much in networking, but in some areas of architectures. If something was not completely clear, I had technical discussions with the program managers. That's one of the nice things about program managers at DARPA; they are very good technical people, but if they could not present something clearly to me or understand something thoroughly, they would pick up the phone and next day they would have the person who wrote the proposal visit us and explain it in some detail. We would have detailed technical interactions in these visits. I got very much involved in some of the technical issues, especially in proposals that seemed to be poor. At times, proposals came to the agency through political links. Somebody had a friend here or there and they got a hearing with the director of the agency and they want you to come to hear their idea. Sometimes, it turns out to be a scientifically unsound idea. It was one of the most difficult things, to try to analyze what is proposed, to dissect it, to see why it is unsound, to convince the director of the agency that this is the case, and for him, then, to convince his friend or congressman that we are not saying no for some arbitrary reason, but because the idea is not technically sound. This took some time. I remember some special cases; there are famed cases. It takes a certain amount of critical attitude, but also a certain amount of diplomacy to not go overboard in these situations. DARPA managers and office directors are well known for going quickly to the heart of the matter, the technical heart of the matter, and shunning boiler plate and all these long introductions and background material. Sometimes people coming for a presentation had the impression that the response that they encountered at DARPA was too abrupt, even rude. If, after two or three minutes of introductions, he didn't get to the heart of the technical proposal or the main idea, usually he would be given a hard time. As soon as he got to the main idea, then things became nicer. People who have been around DARPA for some time are used to that, to this style which looks a little bit too abrupt. You've got to get quickly to the heart of the matter and tell me what is your idea and what's great about it, why it's new and how does it relate to this and the other existing ideas. It is a tough environment for some people and some people are intimidated and don't come again after they've had a difficult experience with some of the program managers, and some of the office directors.

NORBERG: Did you hire any program managers while you were there?

AMAREL: Oh yes. The office was almost completely filled by me.

NORBERG: What sort of qualities did you look for in these people?

AMAREL: Technical soundness first, understanding of the field. Preferably, they themselves have done some research in the area. A certain amount of dedication, a good personality, good judgment. It's very hard to define these things. It's not very different from trying to hire a new faculty member at the not very senior, at the associate professor, level. It's a similar kind of thing.

NORBERG: I would think these people would have to have also a management capability which you wouldn't necessarily expect in a new associate professor.

AMAREL: True.

NORBERG: We're talking about millions of dollars being monitored here and major programs.

AMAREL: They are doing this, but there is a lot of support and a lot of checks and balances throughout. They have financial advisors in the office; there's a financial manager. There are people in the program management office, a separate office of DARPA, with whom they always can interact. They have support personnel and support contractors to do administration. The fundamental requirements for good program managers are not fiscal know-how and administrative know-how, but the ability to make sound judgments about the quality of programs, whether the programs are progressing well, and what should be done to handle unforeseen difficulties in programs. Technical, scientific know-how and judgment about what is really going on in a program is much more important than administrative skills. Of course, you have to fill forms, you have to write reports, to organize meeting, etc. Much of that is done with a lot of help from support contractors and from other financial people. Management, in the sense of research management, is an important function of a program manager. A good research manager should know what the goals of the program are, what everybody is doing. He should understand what the new ideas and difficulties are, get a sense that what is being done to handle difficulties is reasonable, and understand why something is not being done. It is the same thing as being a laboratory manager.

NORBERG: Saul, I'd like to turn to one last topic and allow you a little time to wax philosophical, perhaps.

AMAREL: Oh, boy. I am sure that that is not very important. I can't give you any more coffee!

NORBERG: I want to return to a subject that we also discussed at lunch. This is the question of our case study in this project of AI. I'd like to divide the question that I want to ask you into two parts. First of all, what do you think are important elements in the history of AI as you know it that we should concentrate on (those things associated with the DARPA program, of course) to understand important developments in the field and the influence of DARPA on the field. That's part A. I'll get to part B later.

AMAREL: The work that was done in the general area of problem-solving and planning over the years. It was mostly concentrated, was typically concentrated at CMU and at SRI. It's very important work, very basic fundamental work, it's driving many other things in the discipline. It's worthwhile to review this work, to see what has been done, to relate it to other developments in the field. It was seminal; it was basic. In the area of applications and expert systems, the work through Stanford, because that was the main conduit for the ARPA connection to that particular work, and later on the work in the early stages of Strategic Computing, in knowledge-based systems. That is of enormous impact on what is going on in the country in the deployment of expert systems. Work in vision from the beginning has been always a very important driver; DARPA has been the driver of the work on vision in this country. Understanding visual scenes, not just pattern recognition and matching and so on. Understanding visual scenes. This has been a tremendously seminal activity, lots of very good scientific work was done. MIT was involved as well as other places, Carnegie-Mellon. Ron Ohlander, did some work in the area of vision, himself, in the understanding and of characterization of textures. He did some work himself in texture understanding and so on. Speech and language also were areas of AI that were very much advanced by DARPA.

NORBERG: Is this language understanding now?

AMAREL: Natural language understanding and speech. They were separated for some time. But, again, one of my contributions was to bring these two things together. My feeling is that speech was separated from natural language during Chomsky because he focused very much on the symbolic parts of natural language. But usually linguists always consider speech and the symbolic parts of the language together. I think it makes a lot of sense to study speech and the symbolic part of language together for various reasons. In order to detect and to understand speech, you have to know what is being said, so the symbolic part has to be there to some extent. And the other way around, sometimes in order to understand the symbolic part, some of the speech features such as intonation, the way in which things are being said, can help a lot. It would be nice if you could even see the other person, as we are doing now, nodding and so on. This is part of the understanding of spoken communications. I would like to bring speech and language together, and DARPA has been doing this recently. It was very hard to achieve this sort of thing in a place like NSF or other places, because you need large programs to do that. You need several people working on a project to be able to look at speech from the phonetics level through the lexical level all the way through the semantics and the pragmatics levels and the symbolic ways of processing speech. You need large projects. There are projects in this area that cannot be done in a small size. The same thing with vision. I always felt that what is being done in scientific approaches to vision by NSF and others -- that you take a small corner of the field and you look into that in great detail; you know how to detect edges in the scene, you want to find out certain kinds of visual features in the scene or how to take care of texture, would clarify only a limited part of vision phenomena. In order to do vision, you have to take all of those things together and to go all the way up to the point of having an internal representation of the scene, to try to match objects in the scene with models that you have stored, and to be able to identify objects and relationships of interest in the scene. So, in order to be able to know that you have advanced the field of vision, you must have a problem of some complexity that requires you to go from the physical details of the optics of the scene up to the point that you have internal representations of the scene, and you can control effectively actions that are based on these internal representations. I was always very interested in having vision related to action, to robotic action, to show that indeed you have achieved vision understanding; alternatively to have vision in the context of a question answering system, so that you can ask questions about some scene and you can see that you understand the scene because you can answer questions about the scene in an appropriate manner. In order to do that you need big projects. That's what I believe is a very important thing that DARPA is doing or has been doing, in the discipline. There are some information processing problems that by necessity require not a partition of the field in small pieces but an integration of pieces into a chunk of appropriate, natural size which can be done by several people working together and which involves considerable amounts of experimentation. DARPA knows how to handle this sort of project. It has done it in the area of speech and language, and it is doing it in the area of vision and vision with robotic action now. This is a very important style of doing research. So, big science in computer science is something that DARPA has done and it continues to do. Projects that are not $50K to $100K but they are 1 to 2 million dollars per year. That's the kinds of levels of effort you need with the appropriate computing equipment. Lots of infrastructure to do it, because you need to do experimentation with appropriate computer power.

So, I said that DARPA had strong impact in the AI areas of problem-solving, foundations, applications of problem-solving through knowledge-base systems, speech, spoken language, vision. Another element which may not be just pure AI, but is absolutely essential, and that's what connects AI with other parts of computing, is the system background, the system's infrastructure for doing experiments in AI. The development of the LISP machines, for instance, the development of the interfaces, the development of Common LISP, which was a very important achievement, the push of LISP, the development of shells, the development of software tools, utilities, windows, and environments that allow a person to do experiments, quick experiments, rapid prototyping, in complex situations, these very important contribution of DARPA. It's part, again, of what we discussed before, that is you want to have more emphasis on the design tools for building programs rather than on the power of machines for running programs. This is an important contribution of DARPA. I don't know if it's because you need that for AI, but in the bundle of things that I can put in someplace and say they are contributions of DARPA related to AI, this is one, too. But then you can say that that could also go in other areas of computing and other areas of software development, and I believe so, too. I believe that a very important side-effect of work on AI is the development of a very interesting software development technology that is having an impact in other areas of computing.

NORBERG: That's my part B. Can you cite some examples for me of the influence of AI in other areas of computer science.

AMAREL: This is what I'm just saying. Software development technology is a very important example, and architectures of machines that are especially appropriate for doing interactive program development, windows. Various kinds of debugging tools that came up with InterLisp for instance, various kind of help devices for users of large programs. All of these are small things, but they came up because in the course of developing AI programs researchers wanted to have devices to help explain why certain program features produced certain behaviors. In expert systems there has been much emphasis not only in producing a decision which is based on various facts and various pieces of knowledge, but also in having, on demand, an explanation for the basis of that decision, often in the form of a trace of the entire line of reasoning in support of the decision, that is what pieces of knowledge were used to reach the decision, how, and where. So you have a very clear way of really auditing a decision, a reasoning trail. This is very important. It's important not only for AI; it's important for any other area of computing where you want to really be sure that a computation is sound, not only in an overall input- output way but als o in its details. It's an extension of the idea of debugging. It's a very powerful way of really auditing what goes on in a computation.

So I would say that program development environments and tools, and the entire technology of programming in an expert systems style, which is a different style of programming, are important contribution of AI to computing in general. A language like LISP and all the symbolic manipulation languages go beyond what has been done in imperative languages like FORTRAN and Pascal and so on. LISP is a functional language; Prolog is a logic language; they both grew within the AI context. They provide different ways of doing programming at a high level. The initial dream of LISP was to have something which is almost a mathematical system in which you could manipulate programs. It didn't work out this way. People who worked on Prolog initially thought that pure Prolog is manipulatable logically. It never worked this way, because you always have to balance a capability for reasoning with capabilities for expression. Sometimes you need more expressive power, so you add all sorts of kludges in there and this spoils the initial "pure" intent. But DARPA has done something beautiful with Common LISP. It standardized LISP to a language that could be used on many machines. People could transfer easily programs from one machine to another. Transferability is very important. So, I think the development of symbolic languages, LISP, Prolog and various other languages around them, I think those are very important contributions of the activity in AI, and especially of the activity supported by DARPA. You'll say there are other symbolic manipulation languages. SNOBOL, for instance, at some point, was an important language. It's not around anymore; it's a defunct language. It was developed at Bell Labs. Bell Labs needed a thing like that to do symbolic manipulation in other kinds of situations. It was very much an academic language. I don't know how much it has been used in Bell Labs, or let's say in the Bell companies, in terms of the computing they had to do. It didn't grow with DARPA support. It grew in an environment where people wanted to do analysis of language, of text, parsing, rather than other kinds of symbolic computations. By the way, I want to come back to something that Keith Uncapher and I discussed. I don't want the term symbolic computation to be used instead of AI. I want AI to be called AI. I want AI to be called AI. Symbolic computation is a very broad category of computations. It includes many, many things. As a matter of fact, there are people who say it includes also numerical computation, because numbers, after all, are certain kinds of symbols and what computers do, in general, is symbolic manipulation. I would say it would be much more accurate to talk about numerical computation and non-numerical computation. Under non-numerical computations, I think it would be inappropriate to say that all non-numerical computations are AI, because there are non-numerical computations that have to do with manipulation of programs! For example, when you manipulate programs in a compiler, you have a non-numerical computation. AI is a special kind of information processing activity within mostly non-numerical computations. But there are numerical computations in AI also. For instance, in low level vision, you have tremendous numbers of numerical computations on light intensities. So I don't think it's appropriate to categorize the field of AI along the dimension of numerical, non-numerical or symbolic computation. What we are talking about is something which initially Carnegie-Mellon called complex information processing, and later on McCarthy and everybody in the field kept calling AI, and now it continues to be called AI, for better or for worse. We can debate that. Still it's a special kind of set of activities with a community of workers out there, scientific conferences, and journals and so on, that are identified with AI as a discipline. This is the discipline that DARPA has been supporting. Within DARPA, it changed its name to intelligent systems, machine intelligence, artificial intelligence, but essentially it is AI, not symbolic computation.

NORBERG: Can I ask you one more question, we're almost out of time. Can you think of any examples from datastructures where AI has been influential?

AMAREL: Of course. The entire work on deductive databases. Oh! you're talking about datastructures as such? I mean, the study of algorithms, or databases?

NORBERG: Development of databases.

AMAREL: The entire area of deductive databases, which is a large activity today; it involves the following: how do you answer a query which may be stated in logical language in such a way that you consider all the entries in your database as axioms; if you can answer the query by pure matching, which is the way elementary databases do that, that's fine; if you cannot, then try to find a way of deducing the answer to the question from the information you have in the database. Tremendous amount of activity. It started within AI. It has its own life now. There are lots of theoretical activities within this area that has to do with the complexity of answering different classes of queries. It's one of the areas of AI, if you wish, where there is some interesting theoretical work going on. For instance, a colleague of mine here, Imielimsky, has been working on a problem which, I think, is going to be a great significance in the future within AI and other parts of computer science. Suppose you are asking a question to a deductive database, and in addition you know that you have a finite amount of time within which you have to get an answer. It's very important for you to get some "sensible" answer within that time. Now, you would like the system to be able to respond to you in the following way: it would evaluate how much it would take the system to compute a complete answer and, on basis of that, it would choose a computation that may be an approximate computation that within the given bounds of time would give you an answer that is meaningful, meaningful in the sense of a higher order but in a numerical computation. He would like to obtain something similar but not necessarily in numbers -- some sort of meaningful qualitative answer to a question. Human beings do that all the time. Under certain limitations of time we give a quick summary, and we give different kinds of summaries, under different circumstances. Sometimes we give the main gist of a thing in one sentence or in one word, sometimes we can give a paragraph. Now, from the computational point of view this means that there's no single computation that provides an answer to a question, but there is a partial order of computations, each of them ordered by time complexity, by the degree of computational complexity it takes to obtain them, and you would like to have a problem-solving system, in this case a query answering system, that is able to answer you with a given parameter of time in the best possible way in terms of meaning. Thomas Imielinsky is working on this now; I'm excited about it, I'm starting to work with him on that. It's part of a new area within AI which is called real-time problem-solving. It will take more than raw power of computing to do that. It will take the judgment about what kind of computation you are going to mobilize to do that for you. There are some computations that you know are going to give you simplified, or approximate answers. The question is do how you initially develop this set of computations that provide different degrees of approximation, and how do you decide online which one to choose, given the time constraints.

END OF INTERVIEW