AI Magazine Volume 18 Number 2 (1997) (© AAAI)



# Artificial Intelligence

## What Works and What Doesn't?

*Frederick Hayes-Roth*

■ AI has been well supported by government research and development dollars for decades now, and people are beginning to ask hard questions: What really works? What are the limits? What doesn't work as advertised? What isn't likely to work? What isn't affordable?

This article holds a mirror up to the community, both to provide feedback and stimulate more self-assessment. The significant accomplishments and strengths of the field are highlighted. The research agenda, strategy, and heuristics are reviewed, and a change of course is recommended to improve the field's ability to produce reusable and interoperable components.

I have been invited to assess the status of progress in AI and, specifically, to address the question of what works and what does not. This question is motivated by the belief that progress in the field has been uneven, that many of the objectives have been achieved, but other aspirations remain unfulfilled. I think those of us who've been in the field for some time realize what a challenge it is to apply AI successfully to almost anything. The field is full of useful findings and techniques; however, there are many challenges that people have forecast the field would have resolved or produced solutions to by now that have not been met.

Thus, the goals that I have set for this article are basically to encourage us "to look in the mirror" and do a self-assessment. I have to tell you that I'm in many places right now where people often jest about what a sorry state the field of AI is in or what a failure it was. And I don't think that's true at all. I don't think the people who have these opinions are very well informed, yet there's obviously a germ of truth in all this. I want to talk about some areas where I think the field actually has some problems in the way it goes about doing its work and try to build a shared perception with you about what most of the areas of strength are.

I think there are some new opportunities owing to the fact that we have accomplished a good deal collectively, and the key funding organizations, such as the Defense Advanced Research Projects Agency (DARPA), recognize this. In addition, the Department of Defense (DOD) is increasingly relying on DARPA to produce solutions to some challenging problems that require AI technology. These significant problems create opportunities for today's researchers and practitioners. If I could stimulate you to focus some of your energies on these new problem areas and these new opportunities, I would be satisfied.

At the outset, I want to give a couple of disclaimers. I'm not pretending here to do a comprehensive survey of the field. I actually participated in such an effort recently, the results of which were published in the *Communications of the ACM* (Hayes-Roth and Jacobstein 1994). In that effort, I tried to be objective and comprehensive. In this article, however, I'm going to try to tell you candidly and informally the way it looks to me, and I would entertain disagreement and discussion gladly. However, I think any kind of judgment is value laden. I do have some values. They're not necessarily the same as others, but I think they represent a pretty good cross-section of the viewpoints of many of the people in the field and many of the people who patronize the field (in both senses).

My values are that a field ought to be able to demonstrate incremental progress, not necessarily every day, every week, but over the

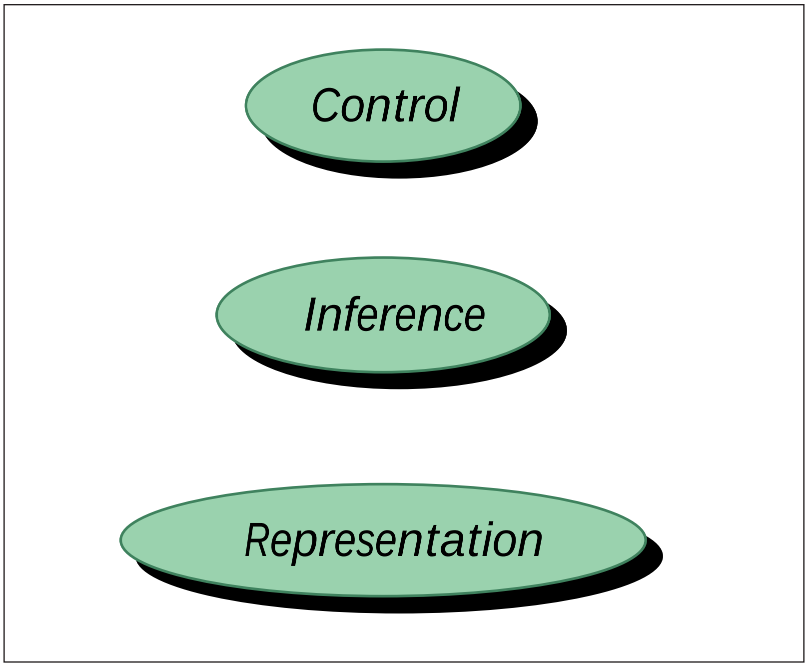


*Figure 1. The Three Basic Elements of Symbolic Reasoning.*

course of years and decades, certainly you ought to be able to demonstrate that you're standing at a higher level than you were years before. I think to a large extent that the field is subsidized as a darling of the domestic and military research and development communities, and we owe them something for all the investment. To the extent to which we can contribute solutions to problems in those areas, I think that's good.

I have been an engineer, I've been a technologist, and I've been a scientist. I think these are different enterprises. *Science*, for me, is discovery of truth from observation. *Engineering* is what we do when we're asked to build something, even perhaps without certainty that the methods we pursue will lead to a successful outcome and perhaps not even knowing why it works when we're done. And *technology* feeds off these other two fields and feeds into them by producing reusable components and processes. In many ways, civilization is an accumulation of these technological advances, and I think we ought to be able to assess where we are in producing these reusable technological elements.

Certainly, no matter which area you're in, you have to be interested in the repeatability of your results and whether they generalize from one situation to another. In addition, as several of the American Association for Artificial Intelligence (AAAI) and Innovative Applications of Artificial Intelligence papers have pointed out, you must be concerned that the techniques we demonstrate in the laboratory carry over and actually produce material benefit; in short, do our techniques scale up effectively?

What are the questions that I want to focus on? I tried to put myself in the position of the people that the conference chairs were asking me to think about. They're largely government research and development people, such as some of the DARPA program managers who come out of the AI community, go to DARPA, and immediately try to defend AI and promote it and keep it vital. And, I tell you, in these environments, people really are asking what they consider to be hard and pointed questions. You would think that these people would have good answers readily available. However, although many people have tried to address such questions, it appears difficult to boil down the answers to these questions.

What do we have that really works, and what are the constraints that determine whether it works or whether it breaks? What are these false advertisements that people contend that the field has been rife with? And are we willing to say that there are certain things that are just beyond the limits of what might be achieved? What are these?

These days, affordability is a big issue. The United States, of course, is in a period of adjustment with a very competitive global environment. Every day, there's a trade-off being made between politically expedient cost reductions and long-term potential payoffs, and affordability is an extremely important concept these days, much more than before. I think all this is an attempt on my part to rephrase the debate because absent this, many people in Washington will maintain the position that, “Well, AI, wasn't that the thing we talked about a decade ago, and if it's not done, didn't it fail?”

To a large extent, AI—at least the symbolic part of AI—is built around the integration of three basic capabilities: How do you represent things so that you can do problem solving on them, the problem solving being produced by inferences drawn by various kinds of engine applied to various kinds of reasoning logic and then, because most problems are exponentially difficult, there has to be some control, and control is often the difference between having something that's usable and having something that's not even interesting, in fact (figure 1).

To apply these simple elements, the field learned quite awhile ago that you had to add something to this, and I've called out here three typical kinds knowledge that have to be added to make this work (figure 2). First, consider *domain knowledge*. What is that about? Most often, it's a description, a model, of the essential features of the otherwise arbitrarily complicated and continuous world. Which things are we going to talk about? Which are we going to pay attention to? How are we going to choose a representation that affords reasonable computation?

The difference between the knowledge an apprentice gets in a basic introductory handbook to a field and that which you get by working in the field for a long time is substantial. Expert systems, for example, grew up on the idea that you could extract from humans who were really proficient in an area their expertise, and if you had this expertise, with almost any reasonable inference mechanism you could implement the expertise as a series of deductive rules of one sort or another. If people were already solving problems more or less consciously and symbolically, control was probably not extremely difficult because humans, of course, don't do exponential computing consciously. And so, with the expertise that we had extracted via knowledge engineering, we could implement a basic problem-solving approach. We would implement the rules required to derive valid answers to given problems.

Finally, there are a number of areas where problems are of such complexity or where the extractable knowledge has limited ability to drive immediately to a solution that we encounter a requirement for a kind of control knowledge. I'll call this *management know-how*. It's knowledge about managing problem-solving resources, knowledge about how to recognize in a problem space which areas might be most fruitful or least promising to pursue first.

As people begin to implement application systems, they take these various knowledge elements and embed them in an application, which here I'm just calling a *problem solver* (figure 3). The problem solver in an integrated application brings all these elements together. The basic elements of control, inference, and representation form the foundation. The three kinds of knowledge are also present. But something has been added to enable us to build an application. Here for lack of a better term, I've labeled this a *problem-solving architecture*, by which I mean a framework for inserting these various elements and having them all plug and play together. We have a lot of problem-solving

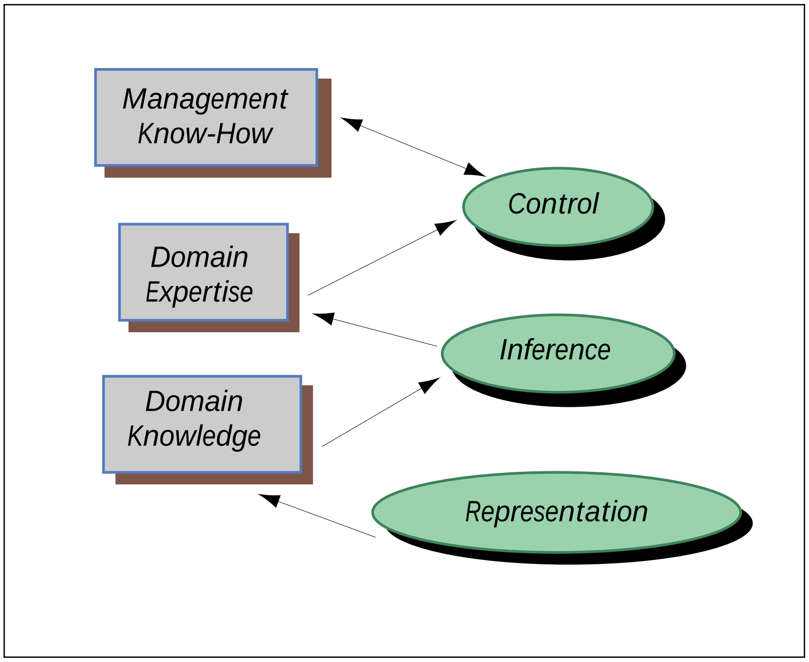


*Figure 2. The Simple Model Must Be Elaborated to Show the Kinds of Knowledge Required to Inform and Use It.*

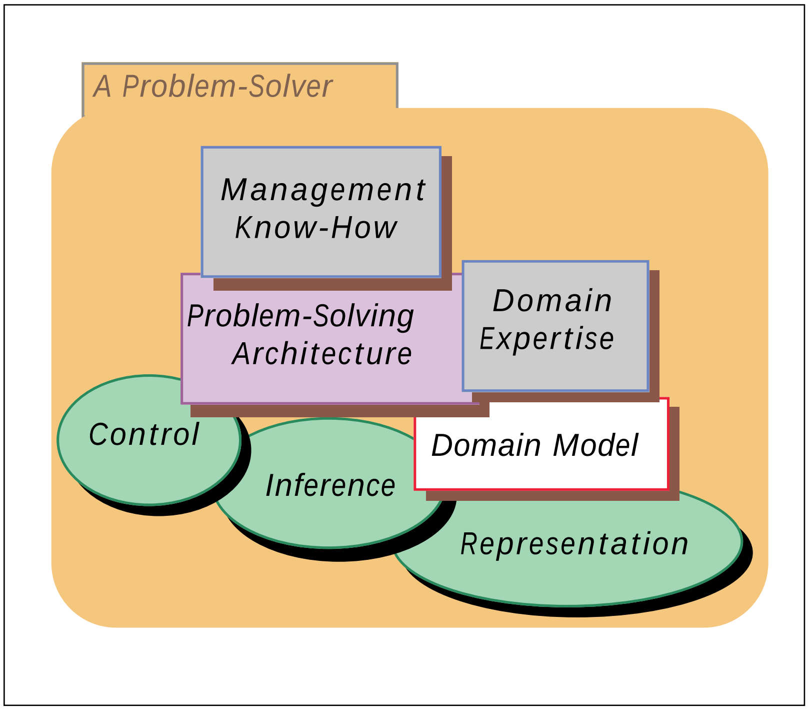


*Figure 3. The Knowledge Must Be Plugged into a Problem-Solving Architecture That Implements Essential Reasoning Capabilities.*

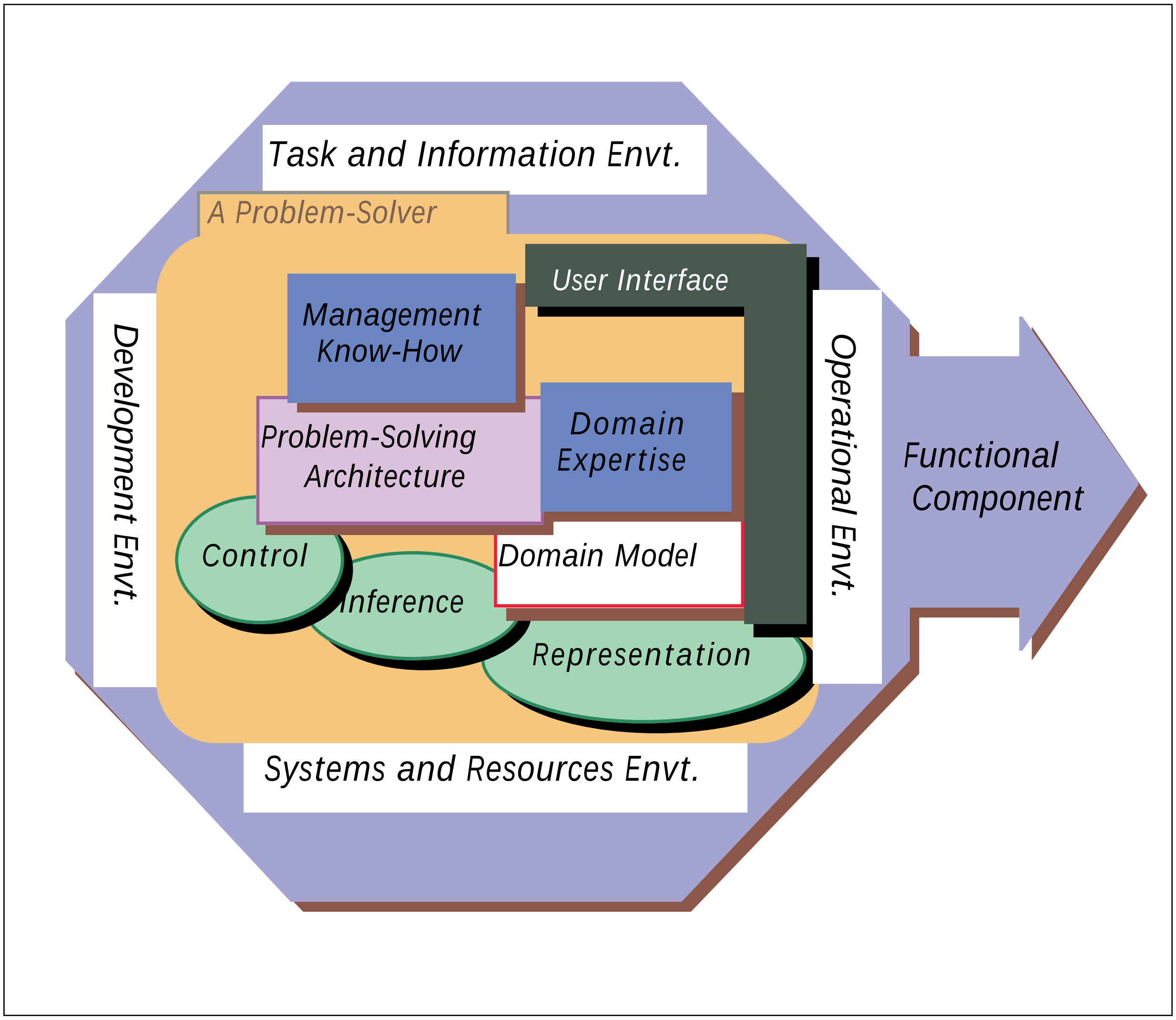


*Figure 4. The Problem Solver Operates in a Context of Several Environments and, Thus, Becomes a Functional Component of a Larger System.*

architectures in the field. The earliest, say, were the back-chaining rule-based systems or the forward-chaining rule-based systems. We have the blackboard systems, case-based reasoning systems, and others.

Each one of these problem-solving architectures turns out to be extremely important to us because today we are not generally able to represent knowledge in a manner that is independent of the implementation milieu. Knowing something about the architecture that we're going to embed our knowledge in enables us to do something that's essential to building an actual system, which is to decide how to shape or mold or craft, if you will—how to phrase the knowledge—so that it interoperates effectively with the other bits of knowledge.

I want to distinguish between the problem solver and what a technologist might call an application or a functional component. The diagram (figure 4) is getting a little complicated; I have to apologize for this. But I want to distinguish here between the inside and the outside. The inside, if you will, is this problem solver. It's the pure embodiment of knowledge to do a task. And now, somehow this task solution is going to be made functional and effective by inserting it into an environment in which it actually adds value. The difference here is between what you might call a pure technical solution—the inside—and what is required for that to be a solution in any organization, which here you might call the membrane, or the outside.

There are actually five different interfaces, if you will, which we don't normally focus on when we're talking about the pure AI technology. And these interfaces, or other environments that we must interface with, actually consume almost all the effort when you try to make the pure technology applicable and effective. The first one, and closest to the problem solver itself, is the *user interface*. Ordinarily, our systems are created to support humans, and we've utilized a range of approaches to make inference accessible, meaningful, and acceptable to the users. Of course, depending on how the user or operator wants to interact with a system, the problem solver can vary from a nearly autonomous system to a nearly manual system, yet the internals need to be approximately the same. There still needs to be a representation of the problem, a representation of the state of the problem solving, and some way to direct control toward the most promising directions. Thus, although the user interface for many researchers has been an afterthought, from the user's point of view, the user interface is the application. It represents a huge percentage of the actual development cost and certainly cannot really be ignored.

Now off in the easterly direction in the figure, toward the right, let me talk about the *operational environment*. Consider, as an example, the problem of scheduling, an area of extensive research in AI. To produce a useful solution, one needs to understand how organizations use schedules. Schedules represent, in the simplest case, agreed-on solutions to task and resource-allocation problems. In other cases, they represent contracts. They represent in other cases actual production-control mechanisms. Each of these uses is very different. You can take the same application and try to put it into different operational settings, and you will find that it will need to be extremely different.

The *task and information environment* turns out to be a major issue because almost all AI applications require access to a lot of information. Getting the information in from its sources has often been referred to as a bottleneck or an integration problem or a knowledge-acquisition problem or a learning problem or a maintenance problem. But the interface between the inside and the outside through this information environment is a major problem and one for which we have relatively poor tools today.

On the flip side, we have the systems and resources that this functional component is going to exploit, or interoperate with. These are often computer systems. They're often network systems today, and to a large extent, the applications that we build today have no model or view of this environment. We do a lot of ad hoc construction to embed or integrate the AI solution in various computing environments; I've seen delays of years to make something work across this barrier.

Finally, on the left edge of the figure, we see the *development environment*, which includes things developers often think about: what are the tools that we have available; what are the languages, compilers, debuggers; and so forth. The field of AI, I think, is afflicted by a major problem here. We used to have the world's best environments. We were working with Lisp environments, for example, Interlisp and the Massachusetts Institute of Technology Lisps and their descendants. Nearly all the ventures to improve and propagate these environments are gone now. The ones that remain are rather marginal in the computing field as a whole. Thus, most of the people who are worried about integrating into these other environments have decided that they have to go with mainstream computing and development technologies, which are really poor. There's no question that technically, this decision represents a huge step backward. We get small footprints, reasonable performance, and lousy development environments. It's a bad bargain.

I want to look at the problem-solving capabilities in a context-free way for a moment. We'll come back to what I might call the context of the problem solver shortly. Table 1 is an effort to summarize what works. I might have omitted a few things, but given my emphasis on symbolic and knowledge-based techniques, this is a pretty good summary of what works. We have basically four sorts of proven technology in the areas of representation, inference, control, and problem-solving architectures. I will allude to each and make some global assessments without attempting to provide a tutorial on the various techniques here.

In representation, we have some reasonably good languages. We have a lot of experience in domain modeling, much more than the rest of the world that's coming to it now. In fact, having done knowledge engineering for a couple of decades, I think we know best what works in this area. I often tell people that I believe that knowledge engineering always works if you satisfy the applicability conditions. That is, if you can find somebody who does a reasoning task, does it repeatedly, and does it well, you can ultimately elicit this

| Table of Contents of Proven AI Techniques | |
|---|---|
| Representation | Languages, Domain Modeling, and Knowledge Engineering<br>Rules, frames, classes, cases, hierarchies, propositions, constraints, demons, certainty factors, fuzzy variables |
| Inference | Theorem-Proving, Heuristic Reasoning, and Matching Techniques<br>Forward and backward-chaining, unification, resolution, inheritance, hypothetical reasoning, constraint propagation, case-based reasoning |
| Control | Goal and data directed, messaging, demons, focus, agenda, triggers, metaplans, scheduling, search algorithms |
| Problem-Solving Architectures | Rule based, object oriented, framre based, constraint based, blackboard, heuristic classification, task-specific shells |

*Table 1. A Concise Summary of Many of the Proven AI Techniques.*

knowledge, represent it, and implement it in a machine. This is an amazing accomplishment, and I do not mean to be hyperbolic in this assessment. In doing such knowledge engineering, we have come with up numerous ways to organize knowledge and make it explicit. These are familiar. They take different forms and different combinations, and they're all useful.

Under inference, the field has demonstrated how to draw valid conclusions, how to do it under resource constraints, and how to do it using uncertain or errorful methods. These methods work, and they're in use on a daily basis in thousands of organizations. I've listed here, among the chaining methods, unification, resolution, and other theorem-proving methods, some of the techniques we employ to simplify knowledge representation and obtain inference for lower costs, such as inheritance or case-based reasoning. Hypothetical reasoning is a somewhat esoteric approach. It's extremely useful to organizations, for example, that have to plan under uncertainty.

In the area of control, AI has invented most of the metaphors and paradigms that will carry computing forward for a significant period of time. We know something about what it means to be oriented in a direction and to organize computing toward some goal or to proceed forward from some data. Messaging is taken for granted today, and in most areas where messaging is being used today by the computing industry as a whole, they have no understanding of what the AI field adopted messaging for in the first place. They haven't gotten to semantics yet. We have developed a variety of ways to develop distributed or cooperative or loosely coupled elements of computing from agenda mechanisms, blackboard mechanisms, and so forth. We have a good handle on the representation of search; we have very good algorithms that, when the "shoe fits," provide an excellent solution. Finally, we have a tremendous amount of experience specializing these algorithms and applying them, for example, to scheduling problems.

In the architecture area, we have techniques for rule-based, object-oriented, frame-based, constraint-based, and blackboard methods as well as heuristic classification and task-specific shells. This body of methods and techniques is a significant accomplishment; so, when I look at the results and summarize what was done over the past few decades, I see that we actually created a large amount of useful and important technology.

These important technology results were falling out, if you will, as a side effect, of conducting research in a certain way. In at least five areas, shown in table 2, the field has focused persistently and with continuous incremental progress. In these areas, we have reached a point where even if we don't have solutions in hand, we have demonstrably usable, if somewhat limited, technology. Let's briefly assess each of these areas.

The first area includes recognition, interpretation, and understanding. Today, you can purchase and employ constrained speech-

| Successes of the AI Research Paradigm | |
|---|---|
| Recognition, interpretation, and understanding | Dozens of constrained speech and message systems |
| Gesture, expression, and communication | Speech generation |
| Analysis and Prescription | Thousands of domain-specific expert systems<br>Hundreds of case-based assistants |
| Planning and Scheduling | A few superb domain-specific systems |
| Integrated behavior | Autonomous and teleoperated vehicles |

*Table 2. A List of Principal Fruitful Research Areas.*

understanding systems. This is a huge breakthrough, one that took nearly 30 years of continuous research and is not done yet. Handwriting, for example, which has been the butt of so much humor in the popular press, is much easier than speech and would have been solved, in my opinion, very effectively by now if it had been pursued continuously. Simply to try and transfer what we know from speech to handwriting is a challenge, but I do not think we're very far away in time from people having commercially credible handwriting systems.

In the area of generation of communication, we're not as far along toward powerful systems for gesture and expression. We have excellent automatic speech-generation systems today. The quality is really superb. I remember vividly the primitive quality of speech synthesis when I visited Bell Labs in the early 1980s. Given the state of the art at this time, we've come incredibly far.

In the area of what you might call analysis or diagnosis and the recommendation or prescription of how to fix problems, we have literally thousands of applications that are in use today. They manifest two basic approaches: The first uses what you might call a strong method, typically based on some expert system shell. In this approach, somebody does some knowledge engineering and figures out how to classify the problem variants, how to sort users' individual situations into one of the set of parameterized categories of problem types, how to pull out a standard recipe for problem response, and how to instantiate the variables in the recipe with the parameters identified during the analysis. This way, many problem variants are treated with the same parameterized solution.

More recent, but in many ways faster-growing, is the case-based approach, where the solution typically uses a more interactive, mixed-initiative system. We still face a requirement to organize the knowledge, but sometimes, knowledge can be accessed by purely syntactic means. For example, when you call a help desk and ask for some help on some problem, the person who's there might have no real understanding but might be typing in what you say as a keyword search to index into a case and a script of how to walk you through a problem-solving session. Although the reasoning here is shallow, the organization and indexing of knowledge is critical.

In the planning and scheduling area, I think it's clear that we've made steady incremental progress. We have some truly outstanding results. One of the most impressive is the work of Doug Smith and his colleagues at the Kestrel Institute in generating schedulers using a software generation technology (Smith, Parra, and Westfold 1996). The refinement approach they use requires an intelligent human software designer to steer a heuristic search method that's using knowledge about algorithm transformations to generate a scheduler. In some recent results, the schedulers that they're producing have literally leapt off the page, being orders of magnitude faster, for real-world problems. Although I think there are other examples in this area, I just wanted to call this one out.

In the area of *integrated behavior,* where we try to link sensing and planning and acting together, we have in some ways just exceeded people's expectations. I omitted entirely when I put table 2 together simple robots, pick and place robots, things like that. I forgot about them because they were an accomplishment of the AI field many years ago.

| Critical Ingredients in the Typical AI Success | |
|---|---|
| Narrow scope | Makes knowledge engineering difficult, hence expensive |
| Focused objective | Assures importance of accomplishment |
| Stability of environment | Allows us to put on blinders and limit concerns |
| High degree of automation & repetition | Gives maximum return for fixed development cost |
| Small project | Less than $50 million, usually much less |
| Lots of custom work by talented professionals | Not cookie-cutter applications |

*Table 3. A List of Heuristics for Successful AI Research to Date.*

Although simple robotic control is in our repertoire, I was thinking more about things that are more autonomous and more capable. In particular I think about the NAVLAB (see www.cs.cmu.edu/afs/cs.cmu.edu/project/alv/member/www/navlab_home_page) example, to cite one, which is the work of Chuck Thorpe and his colleagues at Carnegie Mellon University. We have systems today that, because they're integrated, have taught us the value of looking at the interactions between the way we sense, the way we represent, and the way we plan, which no one would ever have appreciated if they had worked the problems in isolation.

If one does a little analysis of why we're succeeding and then tries to pull out some of the heuristics or situational characteristics that have powered AI's successes, a list of about a half dozen ideas emerges (table 3). First, we generally narrow the problems. The ideal thing to do is narrow them in both dimensions if you can, so they have neither too much breadth nor too much depth. Preferably, the chosen problem will have just enough scope so that if you go through the exercise of implementing a system, it will be valuable. The primary obstacle, of course, is that it is tedious to pull out knowledge and then recode it in a way that machines can apply. I don't think we've made much progress on this problem.

Second, we often decide what we're going to use a system for and then focus narrowly on the objective. The system, for example, will reduce time or reduce cost or produce a better design or something like that. We rarely take on multiple objectives at one time.

Third, if we're wise—and we almost always restrict ourselves to this—we say, "These systems really have to be in fairly stable environments." If we put them in a dynamically changing environment, we're not really sure how to control all the things that we'd have to think about. In any case, to get on with building these systems, we prefer to ignore tangential, environmental, or special case factors; so, we "put on blinders," which is good because it helps us to be successful.

Fourth, we look for situations where there's a high degree of repetition because having made this investment, you'd like to get some payback more than once, which reduces the amount of payback you need each time you use it. If you look at the applications, they tend to be in highly stereotypic environments that afford dozens, hundreds, or even thousands of opportunities to use the same generic solution repetitively.

Fifth, most of these projects are small. I had a lot of trouble deciding what I meant by *small*. I knew it was a project of less than $50 million. Now, for a lot of researchers, that sounds like a lot of money. But I think, as Ed Feigenbaum said, "that's probably the price of a wing on one of these modern aircraft!" The size of the engineering activity that we engage in is truly small scale today. And, in fact, one of the problems that we have when we go into other environments is that we don't have in our culture a methodology or a set of practices that enable us to say, "Oh, having solved small problems we know how to solve big problems." Yet a lot of what people look to AI to do is to augment human intelligence by enabling us to put a lot of understanding and know-how into these systems.

Thus, we're good at small things. And one of the reasons we're good at small things is that we can apply really talented people to them. We do not have many examples today

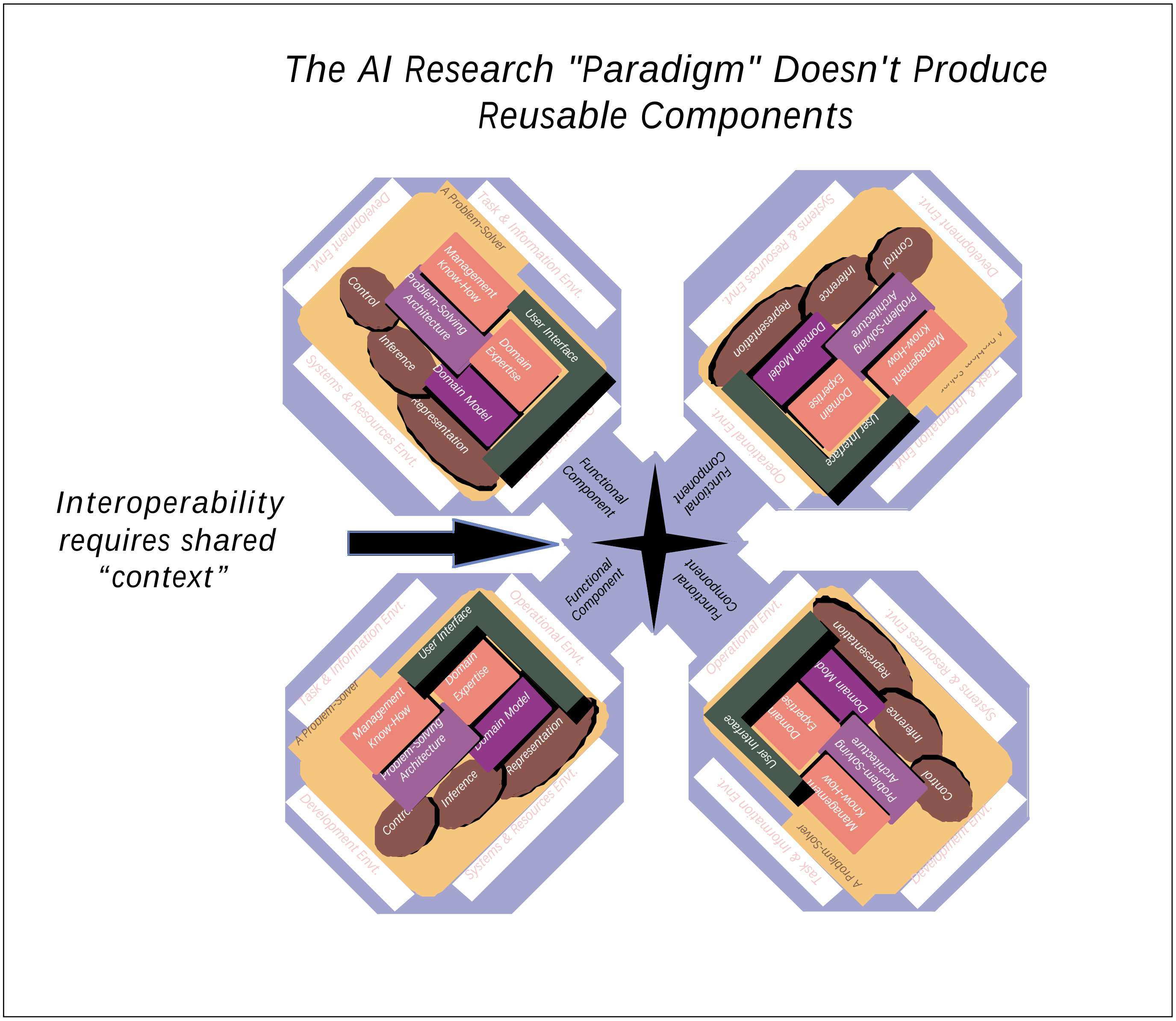


*Figure 5. Each Problem Solver Operates Only within the Context It Was Designed for So That It Cannot Generally Reapply in Another Context or Effectively Interoperate with Other Problem Solvers.*

of what I would call cookie cutter application techniques. If you go to any talk where people relate how they succeed, they'll tell you, "Well, I picked the best technique from each of these many areas, and I adapted it and applied it and I put them all together." In fact, that's how we get these systems to work at all; so, our success basically results from building specific applications today. And each application embeds a kernel of AI capability within a large and thick envelope—a contextual envelope—required to make it function.

Now, one of the problems from this research paradigm is if we take, say, four different functions that have been produced and implemented in their environments, and we try to put them together, they don't interoperate (figure 5). They don't interoperate because even though they might have some knowledge that's the same in each of these areas, they embody so many adaptations that have been made to fit the kernel to the context that it's like combining oranges and apples. Thus, each capability is wrapped in ways appropriate to the application context. Today, for example, we cannot easily use a vision system from one application and combine it with a speech system so that you have a speech system that's improved by having some visual context for what's being referred to. We also cannot combine it with a planner so that it understands some input and some of the contextual variables. We cannot generally lift out the knowledge from one application and use it in another. And I think at its root, this is a major failing of AI. It is too expensive to do knowledge acquisition and knowledge engineering, and we have practically no technology to lift the knowledge out and reapply it.

I think we might be coming to the end of our research paradigm. This paradigm has really worked for us, but it's really problematic, and I don't think it's going to carry us into the next century. It certainly is not going to enable us, I believe, to get to the places where the patrons think the technology is worth spending money on. Let's talk a little about this paradigm. I'm trying to boil down and synthesize what we do. First, we focus on little things, pure things, things that stand alone, things that are context independent. We leave as an exercise to others to figure out how to adapt them and apply them and treat this adaptation as nonessential. In fact, for many students of the field, we reward only the new and the different.

*I think we might be coming to the end of our research paradigm.*

*This paradigm has really worked for us, but it's really problematic, and I don't think it's going to carry us into the next century.*

The second point is that, "Hey! We're just really good scientists." I mean the actual heuristics we're exploiting work well in other research fields. We should not be surprised that they work here too. We have won, to this point, by taking complex problems and dividing them and treating each of the subproblems separately. We specialize within our field in functional subareas and get good at the things we work on. We have noticed, further, that these subarea technologies also need to be specialized by domain of application, and this domain-specific approach provides a tremendous source of leverage. In fact, many of the people who might have been at this AAAI conference are, for example, at SIGGRAPH right now, applying AI to graphics. Finally, like good scientists, we try to adhere to *ceteris paribus* as a heuristic: Keep everything else, other than the essential, constant, so that you can manipulate the key variables. Unfortunately, this approach leads us basically to ignore or control the environment as opposed to study the environment.

Although these heuristics have worked well for us over time, as they have in other sciences, I think they're fundamentally problematic for where we want to go. First, the differences among the various contexts in which we want to embed our applications make it impossible for our applications to interoperate and impede our ability to share and reuse knowledge. That's a significant problem. Second, today, the more useful a system is when we build it, the less general it is. As a consequence, the more useful things are less reusable and adaptable. This problem is significant too. We might be able to reapply some things but not the most useful things. And, conversely, in some sense, the purer the element, the less valuable it is. One of the other consequences is that, today, the measured part of systems that is AI is small, and I don't think it's increasing. And that's a challenge to the survival of AI. In the arena of rapidly evolving technology components, you have to expand or basically vanish.

Thus, to some extent, our successes to date have resulted from selectivity and careful engineering of context requirements. But as I've said, people expect more of us. In fact, people often expect way too much of us, and some of our best representatives have been our greatest problems here. I remarked to myself during the recent chess tournament how amazing it was that we're at a point in the mid-1990s where computer chess is actually reaching the grand-master level. However, I think we're 30 years late on the Newell and Simon prediction that we will have a world chess champion.[1] Now, that's a half-empty and half-full kind of glass. In the case of chess, people are probably not holding our "underachievement" against us. The people who are watching realize what great strides have been made. But I remember a book by Margaret Boden about 10 years ago, where she talked about interviewing ordinary people in an airport in London, and they had definitive expectations of where AI was and would be (Boden 1986). To a large extent, those expectations were that AI, and the world chess champion, would be here by now, and against these expectations, we are late.

What were people expecting we would have by now? Well, we would have intelligent machines. I haven't met one yet. I haven't met one that's even close to it. We would have superhuman capabilities. Well, we have calculators, and we have symbolic expression solvers, and I suppose they're superhuman. We have routers and schedulers, but these are really narrow applications. We'd have machines that we could talk to by now. I mean, the HAL of *2001* is only a few years away! And, in fact, we can speak to machines now with a fair amount of training. I've tried the latest and greatest, and I found it a little bit daunting. Although we've been making continuous progress there, we're late. And, of course, we'd have autonomous machines. I guess that's what some of us mean by agents, and these would be ubiquitous. Now, having all these things in hand, we would be filthy rich as a nation.

The reality is probably almost 180 degrees away from these expectations. All the current feats of AI today have been enormous engineering undertakings. We don't have an enormous field, we don't have billions going into the field; so, the results are limited. The scope of implemented knowledge today is really small. Although we repeatedly demonstrate that the technology works in particular applications, we have little productivity gain from one application to the next. We have very poor technology for evolution of the systems we've built. We have almost no examples of taking independently developed applications, combining them, and producing something that makes incremental progress toward AI.

Thus, the difference between expectations and achievements is really large, and this is a very serious public relations (PR) problem.

I think if we'd had a PR firm, and we'd managed it a little more carefully, we'd probably have a great reputation because we actually deserve an outstanding reputation. You can go through the syllabus, or the product catalog, of contemporary computing and trace a huge fraction of state-of-the-art technologies to seminal and continuous results in AI: development environments, robots, speech, text, distributed computing, data modeling, computer-aided design, and computer-aided engineering—all these are built on these basic elements of representation, inference, and control. The field doesn't get much credit for these contributions because few people are interested in tracing the origins of current technology.

The application of AI within business has been another major success. Hundreds of organizations have implemented thousands of applications using such AI technology elements as expert systems, case-based reasoning, configuration generators, and machine vision. Multimillion dollar returns on investment are not uncommon because many organizations have situations that are ideal for applying this highly specialized technology. They have tasks that occur frequently; are repetitive; have moderate knowledge requirements; and can be operated in a tightly constrained, relatively nondynamic, application environment. In such situations, for example, a system can be built to leverage a service department by reducing errors, reducing time, saving money—that's a formula for commercial success. But, of course, the field aspires to a lot more than such mundane economic successes. In fact, we have achieved even bigger practical successes. For example, in the Persian Gulf War, one AI application used in targeting and scheduling of air flights was judged to be enormously valuable. The DOD determined that this single application produced such significant savings, not to mention its effect on accelerating operations, that it more than compensated for the total DOD investment, through all time, in the field of AI. Thus, in some sense, our balance went positive for a little while.

Continuing our review of successes, we can see positive signs even in the hard problems such as natural language, vision, domain modeling, software generation, planning, and intelligent control. In several of these really hard problems, we are making steady progress, with which you are probably familiar. The fact that we can demonstrate incremental progress against such hard and essential problems is a credit to the ingenuity and perseverance of the researchers in the field. Nevertheless, there is a disparity between what people expect of us and where the technology is today. And we live in a world of increasing budget constraints. As a consequence, I don't think either our paradigm or our environment allows us to keep going the same old way.

*In the long term, AI is essential.*

I recommend a change in the agenda, from the familiar problems to what I might call some twenty-first–century issues. I want to talk about four of these, as listed in table 4. First comes a multifaceted, multiplicity of capabilities, followed by a cooperative role for our technology, affordability, and then manageability.

I think that actually it was my wife, Barbara Hayes-Roth, who coined the phrase *multifaceted intelligence* for some of her applications; so, let me credit her. I think it's an apt phrase. What we want in our systems is something like we expect from people: that they can shift from one task to another task and, in shifting, they don't lose state, and they don't need to stop the world for a few years and recode their knowledge so that it can be reapplied in the new task context. Thus, a multifaceted intelligence presumably would have a bunch of different component capabilities. It would be able to demonstrate contributions in many different functions. It could perform a variety of tasks, and it would, of course, understand a lot more about the world because we would need to

| Critical Ingredients in the Typical AI Success | |
|---|---|
| Multifaceted intelligence | – Comprises many components<br>– Exhibits many functions<br>– Supports many tasks<br>– Understands and knows a lot |
| Cooperation | – Multiple and heterogeneous agents<br>– Comparative advantages<br>– Bandwidth limited<br>– Time stressed |
| Affordability | – Knowledge and component reuse<br>– Knowledge-based knowledge acquisition and update |
| Manageability | – Estimable, predictable, testable development processes |

*Table 4. A Suggested List of New Focus Problems for AI Research.*

communicate with it over a wider range of requirements.

Cooperation actually is something that the field has been pursuing probably for 20 years along a certain set of paths—they're like threads—but I really think it's a tapestry that needs to be produced now. We have multiple and varied agents, and they need to cooperate. Often the agents are people. Often, the people need to share state with machines, and this is dynamic state. This is a real challenge because our ability to express where we are in a problem or to have a machine communicate its state to us is limited by the channels available, the amount of time we're willing to expend, and the bandwidth of our interaction. Usually, in most of the tasks that require cooperation, we are time stressed and are unwilling to convey explicitly all the state information that would be required to align our agents adequately for significant tasks.

I want to call out as the third focus area the problem of affordability. I don't have a lot of simple solutions to address this requirement. The obvious strategy is to reuse stuff, and it's clear that the critical thing to reuse is the knowledge that we put into these systems. To support knowledge reuse, not only do we have to figure out some way to make it be context adaptable, but we also need to apply the technology directly in support of our own processes for acquisition and maintenance.

Finally, it would be nice, going back to the observation that it's a small-project field, to have something like manageability or predictability of the process. We have a very weak process, employing very weak methods. We don't really know whether we're going to be successful when we start a new kind of application, which is a problem.

I think all these points lead to a new paradigm, which I want to try to illustrate. Figure 6 shows an architectural framework, to be defined, which would allow us to integrate four different kinds of knowledge, which could be acquired independently of one another and then applied to support multiple tasks, such as vision and planning and control. Moreover, these tasks, such as vision and planning and control, could be performed in multiple domains, such as on the highways and in a building. This is a goal. If we could achieve this goal, we would, in fact, have a technological basis for sharing that would produce economy and efficiency. As previously discussed, with today's state of the art, knowledge needed to do a function in support of a particular task, in a particular domain, is usually uniquely crafted for any particular system environment. Today, we are far from having the kind of independence of knowledge modules suggested by figure 6.

Thus, I think if we're going to pursue the recommended agenda, we have to have some sustained support for what people might call long-term or large projects. And here I'm addressing the people in government, who are the patrons of a lot of this work. Sustained pursuit of problems is not popular today in government, but only the government is equipped to undertake the implied challenges.

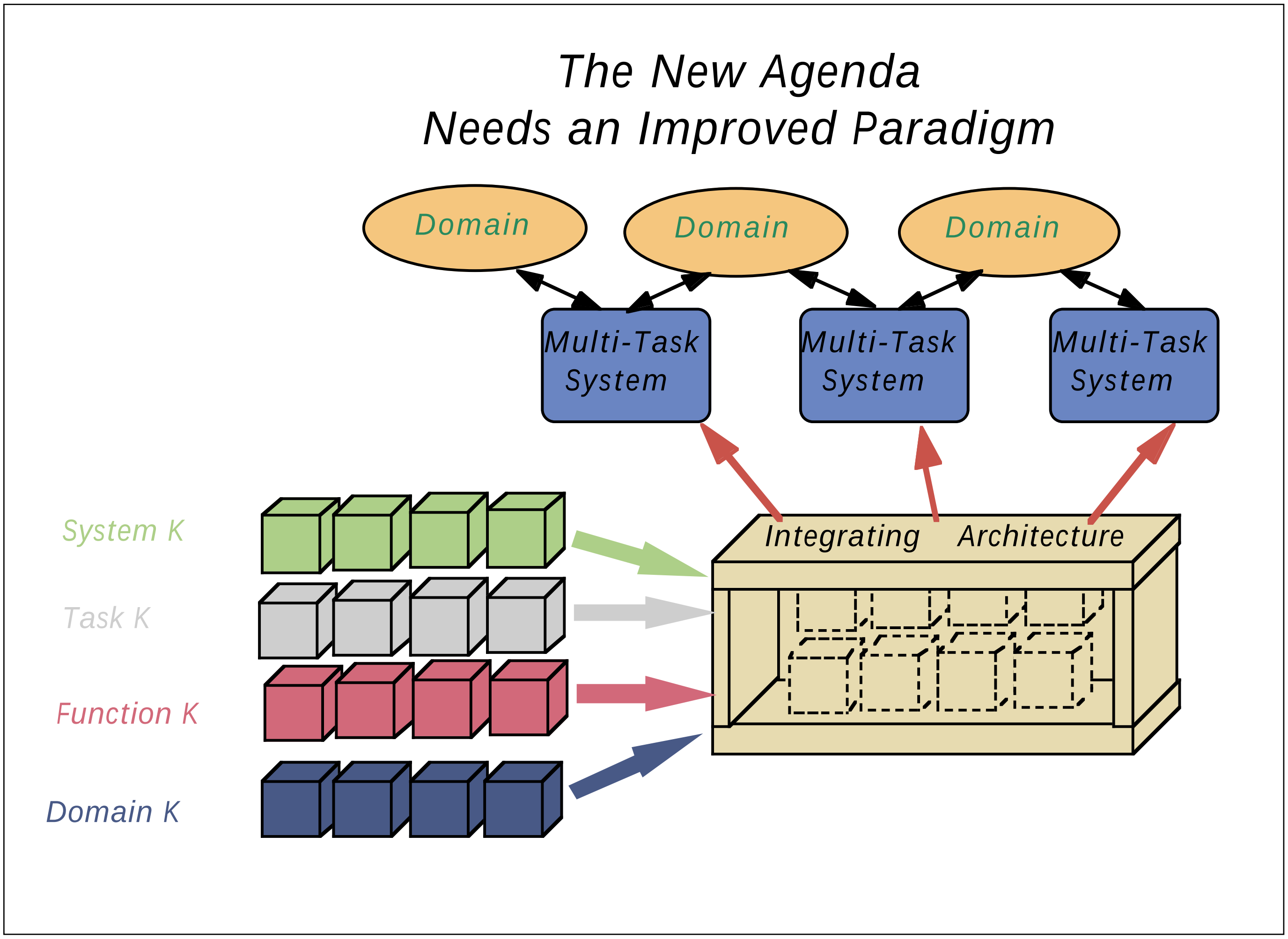


*Figure 6. Future Multifaceted Intelligent Applications Will Need to Combine Several Different Types of Modular and Reusable Knowledge to Be Able to Perform Multiple Tasks in Multiple Contexts.*

What would the recommended strategy be? First, the principal strategy should emphasize context-adaptable building blocks, reusable knowledge, high-value component functions, composition architectures for multitask systems, and an architecture for semantic interoperability. Second, a small number of domains must be attacked persistently. Third, integrated performance across multiple tasks is the first criterion. Fourth, adaptability and transferability are the goals. Fifth, research programs should organize appropriately.

We'd like to have context-adaptable building blocks, as I've previously suggested. We'd like to have some knowledge that we could move from one kind of task and function to another. We would build up a repository of these useful functions. We'd have means and architectures for composing these and producing adapted, situated systems that could do multiple tasks. Of course, underpinning all this is some ability for these components to accord with their contextual assumption or be semantically interoperable.

How are we going to get there? Well, we have to limit the space, and I've recommended that we pick a couple of domains; stay in these domains for a long time; and across these domains, pursue multiple tasks. What we're aiming at is the ability to adapt and transfer our results to new situations.

In the long term, AI is essential. I would not want to be part of any nation that didn't lead in this area. It will, in fact, have dramatic benefits, for economic reasons and for national security. But AI is not just a set of little, narrow applications; this won't get us there. We must find a way to integrate across tasks. We also have to recognize that for the foreseeable future, almost all AI will be in support of human organizations. And cooperation

| Proposed Measures of Progress | Some Potential Key Milestones |
|---|---|
| • Systems do more than one task.<br>• Systems are reapplicable to new domains.<br>– Knowledge adaptation is simplified by powerful tools.<br>• Systems use the same knowledge for multiple functions.<br>– The knowledge is economically acquired, represented, and maintained.<br>• Systems can accept, delegate, and share tasks with others.<br>• Systems explicitly model their environment and adapt to variations in it.<br>• Systems can adapt their approach when constrained by time or resources. | • Standardized, reusable representations<br>• Reusable models and knowledge bases<br>• Multifunction knowledge systems<br>• Multitask knowledge systems<br>• Rapidly adaptable and adapting knowledge bases<br>• Associate systems that can<br>– Quickly learn their "master's" values<br>– Correctly model their "master's" context and state<br>– Intelligently execute a variety of delegated tasks<br>– Extend easily to new tasks |

*Table 5. Elements of a Proposed New Strategy and Approach to AI Research.*

between humans and machines is a different kind of problem than the field has traditionally addressed. Humans are pretty good with ambiguity, scruffiness; they are the only ones who are trusted with values and key decisions; and it would be great if they could delegate a fair amount of work. However, today, this is extremely rare. Most AI is about solving problems, which, if they're successfully done, would allow us to take the corresponding task off the typical human agenda. It would be a new milestone in computer science: systems that simplify busy humans' lives.

One of the problems is that when you go into rich-context, multiple-task operating environments, we don't have the time to explain to our machines all the implicit and explicit values of organizations that are at work, an understanding of which makes the difference between good decisions and poor decisions. I also do not think it's going to be a fruitful area for us to invest in.

How would we know we were making some progress on this line? See table 5. I'd like to have some systems do more than one task, and we have a few early examples. I'd like to have some systems that were applicable to new domains. I'm not aware of any today. Today, reapplication technology is limited to tool kits that people use to build multiple systems.

We'd like to have some systems that use the same knowledge for multiple functions. Again, I'm not aware of any. It would be nice to have these systems delegate, accept tasks, and exhibit cooperation. It's essential for these systems to have a model of their environment so they can adapt to it. And systems that adapt their approach to the resources available and the time available would also be hallmarks of this new approach.

Some of the milestones that I'm looking for include standardized, reusable representations, which would be domain independent and adaptable to multiple domains; reusable models for multiple tasks and domains; reusable knowledge bases; some multiple-function systems, or systems that do multiple tasks; and some ways to adapt a knowledge base to a new environment or a changed environment rapidly. Finally, I would aspire to have associate systems where we demonstrate that we actually can communicate and transfer enough problem state to rely on the correct interpretation and value-added cooperation.

I recently reviewed the work that DARPA is doing and sponsoring in AI. I refer people to their web site (see yorktown.dc.isx.com/iso/planning/ index.html). It has a number of projects that I think are exemplary, and I'm asking people to compliment it and encourage it to go on. One in particular that hasn't quite started yet is this joint forces air component commander (JFACC) AFTER NEXT. It's a system to support planning, situation assessment, real-time reasoning, and multiple functions and multiple tasks. This kind of under-taking will drive research and technology in the directions recommended here, toward reusable and component knowledge, task modules, human-machine cooperation, and multiple domains of applicability.

This brings me to the conclusion. I think that AI has accomplished truly amazing results in less than four decades. However, we're far from meeting the expectations that people have for us. I think that the next decades present us with new challenges. We don't really have the technological foundation to meet these today. Finally, I think that when one looks at the areas where we've made major progress, you can see that continuous and incremental progress actually is both possible and probably sufficient. From my point of view, it matters little if the grand enterprise takes a few decades more, so long as we achieve and demonstrate incremental progress.

## Note

1. As of Sunday 11 May 1997, Herbert Simon's prediction of an AI world chess champion was fulfilled. Although he had predicted the achievement would take only 10 years and it actually took nearly 40, the significance of the result cannot be overlooked. At this point, it is reasonable to assume that we can develop world-beating AI capabilities for any well-defined objective, if only we will stay the course, commit appropriate resources, and hunger to succeed! We cannot do everything, but the field's potential is powerful enough to do practically anything we choose.

## Bibliography

Boden, M. 1986. *Artificial Intelligence and Natural Man.* Cambridge, Mass.: MIT Press.

Hayes-Roth, B. 1991a. An Integrated Architecture for Intelligent Agents. *SIGART Bulletin* 2:79–81.

Hayes-Roth, B. 1991b. Making Intelligent Systems Adaptive. In *Architectures for Intelligence,* ed. K. VanLehn, 301–321. Hillsdale, N.J.: Lawrence Erlbaum.

Hayes-Roth, F. 1995. The JTF ATD Architecture for Agile, Mobile, Collaborative Crisis Response: A Case Study in Architecture-Driven Development. In Proceedings of the First International Workshop on Software Architecture, ed. D. Garlan, 116–126. Pittsburgh, Pa.: Carnegie-Mellon University.

Hayes-Roth, F. 1994. Architecture-Based Acquisition and Development of Software: Guidelines and Recommendations from the ARPA Domain-Specific Software Architecture (DSSA) Program, DSSA-94-01, Teknowledge Federal Systems, Palo Alto, California.

Hayes-Roth, F. 1993. The Evolution of Commercial AI Tools: The First Decade. *International Journal of Artificial Intelligence Tools* 2(1): 1–15.

Hayes-Roth, F., and Jacobstein, N. 1994. The State of Knowledge-Based Systems. *Communications of the ACM* 37:27–39.

Hayes-Roth, F., and Schill, J. 1995. Common Strategy Promises Faster Implementation of Computing Advances. *SIGNAL* (October).

Smith, D.; Parra, E.; Westfold, S. 1996. Synthesis of Planning and Scheduling Software. In *Advanced Planning Technology*, ed. A. Tate. Menlo Park, Calif.: AAAI Press.

Rosenblatt, J., and Thorpe, C. 1995. Combining Multiple Goals in a Behavior-Based Architecture. Presented at the 1995 International Conference on Intelligent Robots and Systems (IROS), 7–9 August, Pittsburgh, Pennsylvania.

**Frederick Hayes-Roth** is chairman and chief executive officer of Teknowledge, where he has worked for 15 years since cofounding the company. Currently, he spends most of his time as chief architect for the Defense Advanced Research Projects Agency (DARPA) Joint Task Force Advanced Technology Demonstration and several other programs and projects related to this. He has focused on the architecture and implementation of distributed intelligent systems for almost two decades. His work along these lines includes a series of seminal and successful efforts, such as the HEARSAY-II speech understanding system and its blackboard architecture, opportunistic planning and metacontrol, distributed module-oriented programming supported by the ABE development environment, the real-time architecture for the PILOTS'ASSOCIATE, the DICAM architecture for distributed intelligent control, and the DARPA ISO reference architecture.

Prior to cofounding Teknowledge in 1981, he held faculty appointments at the Massachusetts Institute of Technology, Stanford University, and Carnegie-Mellon University. He directed the research program in information-processing systems at the Rand Corporation. He was a principal designer of HEARSAY-II, one of the first 1000-word continuous speech-understanding systems; the ROSIE system for programming knowledge systems; and many knowledge systems for government decision making. He edited *Pattern-Directed Inference Systems,* a reference on rule-based programming, and *Building Expert Systems,* a text on knowledge engineering. In addition, he has published numerous articles in such journals as *Spectrum, Computer, Communications of the ACM, Computing Surveys, IEEE Transactions, Artificial Intelligence, Cognitive Science, AFCEA Signal,* and *Systems & Software.*

He has served on the editorial boards of the *IEEE Spectrum, IEEE Transactions on Knowledge and Data Engineering,* and *Expert Systems: the International Journal of Knowledge Engineering.* He received his M.S. in computer and communications sciences and his Ph.D. in mathematical psychology from the University of Michigan. He is a fellow of the American Association for Artificial Intelligence and a member of DARPA's ISAT review board and has published more than 100 technical papers.