Challenge Problems for Artificial Intelligence*

(panel statement)

Bart Selman
(moderator)
AT&T

Rodney A. Brooks
MIT

Thomas Dean
Brown University

Eric Horvitz
Microsoft

Tom M. Mitchell
CMU

Nils J. Nilsson
Stanford University

Introduction: Bart Selman

AI textbooks and papers often discuss the big questions, such as "how to reason with uncertainty", "how to reason efficiently", or "how to improve performance through learning." It is more difficult, however, to find descriptions of concrete problems or challenges that are still ambitious and interesting, yet not so open-ended.

The goal of this panel is to formulate a set of such challenge problems for the field. Each panelist was asked to formulate one or more challenges. The emphasis is on problems for which there is a good chance that they will be resolved within the next five to ten years.

A good example of the potential benefit of a concrete AI challenge problem is the recent success of Deep Blue. Deep Blue is the result of a research effort focused on a single problem: develop a program to defeat the world chess champion. Although Deep Blue has not yet quite achieved this goal, it played a remarkably strong game against Kasparov in the recent ACM Chess Challenge Match.

A key lesson we learn from Deep Blue's strength is that efficient brute-force search can be much more effective than sophisticated, heuristically guided search. In fact, brute-force was so successful that it led Kasparov to exclaim "I could feel — I could smell — a new kind of intelligence across the table." (Kasparov 1996)

The experience with Deep Blue shows that a good challenge problem can focus research, lead to concrete progress, and bring us important new insights. Many AI researchers may not like the particular lesson about the value of brute-force over more "intelligent" forms of search, but, nevertheless, it is a very tangible result. In fact, the issue of general purpose ultra-fast search procedures versus heuristically guided domain-dependent methods is currently being revisited in the search and reasoning community.

Finally, as a meta-issue, we will consider how to measure progress in the field. Determining whether a particular piece of work in AI actually brings us any closer to the ultimate goals of AI has proven to be quite difficult. By introducing a set of well-defined challenge problems, we hope that this panel will help provide some benchmarks against which we can measure research progress.

Eight Challenges for Artificial Intelligence: Rodney Brooks

There are two very different sorts of challenges that I see for Artificial Intelligence — first, our systems are pathetic compared to biological systems, along many dimensions, and secondly, moderately good performance from some approaches has sociologically led to winner-take-all trends in research where other promising lines of research have been snuffed out too soon.

*Appears in Proceedings of AAAI-96, National Conference on Artificial Intelligence, August 1996. AAAI Press, Menlo Park CA.

If we compare either software systems, or robotic systems to biological systems, we find that our creations are incredibly fragile by comparison. Below I pose some challenges that are aimed at narrowing this distance. The challenges themselves are general in nature — they do not solve particular problems, but the acts of meeting these challenges will force the creation of new general purpose techniques and tools which will allow us to solve more particular problems.

Challenge 1. Biological systems can adapt to new environments — not perfectly, they die in some environments, but often they can adapt. Currently our programs are very brittle, and certainly a program compiled for one architecture cannot run on another architecture. Can we build a program which can install itself and run itself on an unknown architecture? This sounds very difficult. How about a program which can probe an unknown architecture from a known machine and reconfigure a version of itself to run on the unknown machine? Still rather difficult, so perhaps we have to work up to this by making some "blocks worlds" artificial architectures where we can do this. This might lead to some considerations of how future architectures might be designed so that software is self-configurable, and then even perhaps self-optimizing.

Challenge 2. Minsky (1967) was foundational in establishing the theory of computation, but after Hartmanis (1971) there has been a fixation with asymptotic complexity. In reality lots of problems we face in building real AI systems do not get out of hand in terms of the size of problems for individual modules — in particular with behavior-based systems most of the submodules need only deal with bounded size problems. There are other ways theory could have gone. For instance one might try to come up with a theory of computation based on how much divergence there might be in programs given a one bit error in either the program or data representation. If theory were based on this fundamental concern we might start to understand how to make programs more robust.

Challenge 3. Recent work with evolutionary system has produced some tantalizing spectacular results, e.g., Sims (1994). But it is hard to know how to take things from successes and apply them to new problems. We do not have the equivalent of the Perceptron book (Minsky and Papert 1969) for evolutionary systems.[1] We need such a new book of mathematics so that we understand the strengths and weaknesses of this exciting new approach.

[1]Note that these evolutionary systems are much more that straight genetic algorithms as there is both a variable length genotype and a morphogenesis phase that produces a distinctly different phenotype.

Challenge 4. We have been living with the basic formalizations made by McCulloch and Pitts (1943) for over fifty years now. Their formalization included that the activity of the neuron is an "all-or-none" process, that a certain fixed number of synapses must be excited within the period of latent addition in order to excite a neuron at any time, and this number is independent of the synapses' previous activity and position on the neuron, that the only significant delay within the nervous system is synaptic delay, that the activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time, and that the structure of the net does not change with time. With the addition of changing synaptic weights by Hebb (1949) we pretty much have the modern computational model of neurons used by most researchers. With 50 years of additional neuroscience, we now know that there is much more to real neurons. Can newer models provide us with new computational tools, and will they lead to new insights to challenge the learning capabilities that we see in biological learning?

Over time we become trapped in our shared visions of appropriate ways to tackle problems, and even more trapped by our funding sources where we must constantly justify ourselves by making incremental progress. Sometimes it is worthwhile stepping back and taking an entirely new (or perhaps very old) look at some problems and to think about solving them in new ways. This takes courage as we may be leading ourselves into different sorts of solutions that will for many years have poorer performance than existing solutions. With years of perseverance we may be able to overcome initial problems with the new approaches and eventually leapfrog to better performance. Or we may turn out to be totally wrong. That is where the courage comes in.

Challenge 5. Despite some early misgivings (Selfridge 1956) back when chess playing programs had search trees only two deep (Newell et al. 1958), our modern chess programs completely rely on deep search trees and play chess not at all like humans. Can we build a program that plays chess in the way that a human plays? If we could, then perhaps we could prove how good it was by getting it to play GO—tree search just cannot cut it with GO.

Challenge 6. All of the competitive speech understanding systems today use hidden Markov models. While trainable, these systems have some unfortunate properties. They have much higher error rates than we might desire, they require some restriction in domain, and they are often inordinately sensitive to the choice of microphone. It seems doubtful that people use HMM's internally (even if one doesn't believe that generative grammars are the right approach either). Can we build a speech understanding system that is based on very different principles?

Challenge 7. We live in an environment where we make extensive use of non-speech sound cues. There has been very little work on noise understanding. Can we build interesting noise understanding systems?

Challenge 8. Can we build a system by evolution that is better at a non-trivial task than anything that has been built by hand?

Integrating Theory and Practice in Planning: Thomas Dean

I issue a challenge to theorists, experimentalists, and practitioners alike to raise the level of expectation for collaborative scientific research in planning. Niels Bohr was first and foremost a theoretical physicist. Ernest Rutherford was first and foremost an experimentalist. It can be argued that neither Bohr nor Rutherford would have made as significant contributions to nuclear physics without an appreciation and understanding of one another's results. By analogy, I believe that a deeper understanding of planning problems is possible only through the concerted efforts of theorists, experimentalists, and practitioners. The current interplay between these groups (perhaps factions is the appropriate word) is minimal.

The pendulum of popular opinion swings back and forth between theory and practice. At different times, experimentalist have had to pepper their papers with equations and theorists have had to build systems and run experiments in order to get published. With the exception of the rare person gifted as mathematician and hacker, the requirement of both theory and practice in every result is a difficult one to satisfy; difficult and, I believe, unnecessary. This requirement implies that every publishable result must put forth a theoretical argument and then verify it both analytically and experimentally. The requirement tends to downplay the need for a community effort to weave together a rich tapestry of ideas and results.

A scientific field can nurture those who lean heavily toward theory or practice as long as the individuals direct their research to contribute to problems of common interest. Of course, the field must identify the problems that it considers worthy of emphasis and marshal its forces accordingly. I suggest planning as such a problem and the study of propositional STRIPS planning a good starting point. By analogy to physics in the 1930's, I recommend continued study of the STRIPS planning (the hydrogen atom of planning) and its stochastic counterparts (Markov decision problems) in parallel with investigations into a wide range of more expressive languages for specifying planning problems (the whole of the periodic table).

In terms of concrete proposals, I mention approaches from theoretical computer science that provide alternatives to the standard measures of performance, specifically asymptotic worst-case analysis. As an alternative to worst-case analysis relying on all-powerful, all-knowing adversaries, I discuss the average-case performance measures and the properties of distributions governing the generation of problem instances. Regarding asymptotic arguments, I consider sharp-threshold functions, the relevance of phase-transition phenomena, and the statistical properties of graphs of small order. The resulting perspective emphasizes particular problem instances and specific algorithms rather than problem classes and complexity results that pertain to all algorithms of a given order of growth. I also point out the embarrassing lack of 'real' planning problems, posit the reason for such a deficit, and suggest how recent progress in learning theory might provide a rich source of planning problems.[2]

Decisions, Uncertainty and Intelligence: Eric Horvitz

To be successful in realistic environments, reasoning systems must identify and implement effective actions in the face of inescapable incompleteness in their knowledge about the world. AI investigators have long realized the crucial role that methods for handling incompleteness and uncertainty must play in intelligence. Although we have made significant gains in learning and decision making under uncertainty, difficult challenges remain to be tackled.

Challenge: Creating Situated Autonomous Decision Systems

A key challenge for AI investigators is the development of comprehensive autonomous decision-making systems that are *situated* in dynamic environments over extended periods of time, and that are entrusted with handling varied, complex tasks. Such robust decision systems need the ability to process streams of events over time, and to continue, over their lifetimes, to pursue actions with the greatest expected utility.

Mounting a response to this broad challenge immediately highlights several difficult subproblems, each of which may be viewed as a critical challenge in itself. Pursuing solutions to these subproblems will bring us closer to being able to field a spectrum of application-specific challenges such as developing robotic systems

[2]Additional details can be found in ftp://www.cs.brown.edu/u/tld/postscript/DeanetalAAAI-96.ps.

that are given the run of our homes, tractable medical decision-making associates that span broad areas of medicine, automated apprentices for helping people with scientific exploration, ideal resource management in multimedia systems, and intelligent user interfaces that employ rich models of user intentions and can engage in effective dialogue with people.

To address the broad challenge, we need to consider key phases of automated decision making, including the steps of perceiving states of the world, framing decisions, performing inference to compute beliefs about the world, making observations, and, most importantly, identifying a best set of actions. Under limited resources, we also need to carefully guide the allocation of resources to the different phases of analysis, and to extend decision making to the realm of monitoring and control of the entire decision-making process. I will dive into several problems associated with these components of decision making.

Subproblem: Automated Framing of Decision Problems

Faced with a challenge, a decision-making system must rely on some rules or, more generally, a model that expresses relationships among observations, states of the world, and system actions. Several methods have been studied for dynamically building representations of the world that are custom-tailored to perceived challenges. *Framing* a decision problem refers to identifying a set of relevant distinctions and relationships, at the appropriate level of detail, and weaving together a decision model. Framing a decision problem has resisted formalization. Nevertheless, strides have been made on model-construction techniques, typically relying on the use of logical or decision-theoretic procedures to piece together or prune away distinctions, as a function of the state of the world, yielding manageable focused models. We have a long way to go in our understanding of principles for tractably determining what distinctions and dependencies will be relevant given a situation.

Subproblem: Handling Time, Synchronicity, and Streams of Events

Autonomous systems must make decisions in an evolving environment that may change dramatically over time, partly in response to actions that a system has taken or will take. Most research on action under uncertainty has focused on models and inference procedures that are fundamentally atemporal, or that encode temporal distinctions as static variables. We must endow systems with the ability to represent and reason about the time-dependent dynamics of belief and action, including such critical notions as the persistence and dynamics of world states. We also need to develop better means of synchronizing an agent's perceptions, inference, and actions with important events in the world.

Subproblem: Modeling Preferences and Utility

The axioms of utility give us the fundamental *principle of maximum expected utility*: an agent should take actions that maximize its expected (or average) measure of reward. Although it is easy to state the principle, we are forced in practice to wrestle with several difficult problems. Where does information about the utility of states come from? Whose utility is being maximized? How can we derive utilities associated with solving subproblems from assertions about high-level goals (e.g., "survive for as long as possible!") or from utilities on goal states? What is the most reasonable utility model for evaluating a finite sequence of actions an agent might take over time (e.g., should we assume an infinite number of future actions and discount value of future rewards, or assume a finite set of steps and compute average reward?, etc.). Different assumptions about the specific structure of the utility model lead to different notions of the "best" behaviors and to different computational efficiencies with evaluating sequences of plans.

Subproblem: Mastery of Attention and Architecture

Perceiving, reasoning, and acting all require costly resources. Controlling the allocation of computational resources can be a critical issue in maximizing the value of a situated system's behavior. What aspects of a problem and problem-solving strategy should a system attend to and when? We need to develop richer models of attention. There is promise in continuing work that turns the analytic machinery of decision-theoretic inference onto problem solving itself, and using such measures as the expected value of computation (EVC) to make design-time and run-time decisions about the ideal quantities of computation and memory to allocate to alternative phases of reasoning — including to the control processes themselves. More generally, there is great opportunity in applying these methods in offline and on-line settings to optimize the overall nature and configuration of a system's architecture, including decisions about the compilation of results.

Subproblem: Learning about Self and Environment

Continual learning about the environment and about the efficacy of problem solving is critical for systems situated in complex, dynamic environments, especially when systems may wander into one of several specialized environmental niches. We need to better understand how we can endow our systems with awareness of having adequate or inadequate knowledge about specific types of problems so that they can allocate appropriate resources for exploration and active learning.

There has been research on methods for computing the confidence in results given a model and problem instance. This work highlights opportunities for developing methods that an agent could use to probe for critical gaps in its knowledge about the world. Continuing this research will be valuable for building decision-making systems that can perform active, directed learning.

Subproblem: Living Life Fully—Harnessing Every Second

To date, most of our reasoning systems have no choice but to idle away the precious time between their active problem-solving sessions. Systems immersed in complex environments should always have something to do with their time. We need to develop techniques that allow an agent to continuously partition its time not only among several phases of a pressing analysis but also among a variety of tasks that will help the agent to maximize the expected utility of its behavior over its entire lifetime. Tasks that can benefit from ongoing attention include planning for future challenges, probing and refining a utility model, prefetching information that is likely to be important, compilation of portions of expected forthcoming analyses, experimenting (playing?) with its reasoning and motion control systems, and learning about critical aspects of the external world.

Think Big — AI Needs More Than Incremental Progress: Tom Mitchell

1. Let's build programs that turn the WWWeb into the world's largest knowledge-base. Doug Lenat had a good idea that we should have a large AI knowledge base covering much of human knowledge. One problem with building it is that it might take millions of people. And then we'd have to maintain it afterwards. The web is built and growing already, and it's online, and people are already maintaining it. Unfortunately, it's in text and images and sounds, not logic. So the challenge is to build programs that can "read" the web and turn it into, say, a frame-based symbolic representation that mirrors the content of the web. It's hard, but there is no evidence that it's impossible. And, even partial solutions will be of incredible value.

2. Apply Machine Learning to learn to understand Natural Language. Natural language has always been considered to be too difficult to do for real. But things have changed over the past three years in an interesting way — for the first time in history we have hundreds of millions of supervised training examples indicating the meaning of sentences and phrases: those hyperlinks in all those web pages. Now agreed, they're not quite the kind of supervised training data we'd ask for if we were to choose training data to learn natural language. But when it comes to training data you take what you can get, especially if it numbers in the millions (when there are less than 100,000 words to begin with). So each hyperlink like `my recent publications` has a meaning that is revealed by the web page you get if you click on it. How can we learn something useful about natural language understanding from this kind of data? (We probably need to use more than just this kind of data to learn language, but once we have some basic ontology defined, this kind of data should be of great use in learning the details.)

3. Let's build agents that exhibit *life-long* machine learning, rather than machine learning algorithms that learn one thing and then get rebooted. Consider people and consider current ML approaches such as decision tree or neural network learning. People *mature*, they learn things, then use these things they know to make it easier to learn new things. They use the things they know to choose what to learn next. ML has made good progress on approximating isolated functions from examples of their input/output. But this is just a *subroutine* for learning, and it's more of less a solved problem at this point. The next question is how can we build agents that exhibit long-term learning that is cumulative in a way more like people learn.

Toward Flexible and Robust Robots: Nils J. Nilsson

I start with the premise that it would be desirable to have mobile, AI-style robots that have *continuous existence* — ones that endure and perform useful work for long periods of time rather than ones that merely hold together long enough for a quick demo and video-taping session. Of course, some limited-capability robots — such as those currently used in automobile assembly and hospital-item delivery — do stay on the job for long periods of time. But all of these robots are far from being as robust and flexible as we want robots to be.

My challenge problem is to produce a robot factotum and errand-runner for a typical office building — an office building that is not specially equipped to accommodate robots. The kinds of tasks that such a robot will be able to perform will depend, of course, on its effectors and sensors as well as on its software. There are plenty of important challenge problems concerned with sensors and effectors, but I leave it to others to pose those. Instead, my challenge is to AI people to develop the software for a robot with more-or-less state-of-the art range-finding and vision sensors and locomotion and manipulation effectors. Let's assume that our robot can travel anywhere in the building —

down hallways, into open offices, and up and down elevators (but perhaps not escalators). Assume that it can pick up, carry, and put down small parcels, such as books and packages. For communication with humans, suppose it has a speech synthesizer and word or phrase recognizer and a small keyboard/display console. Any such set of sensors and effectors would be sufficient for an extensive list of tasks *if* only we had the software to perform them.

Here is the specific challenge: The robot must be able to perform (or learn to perform with instruction and training — but without explicit post-factory computer programming) *any task* that a human might "reasonably" expect it to be able to perform given its effector/sensor suite. Of course, some tasks will be impossible — it cannot climb ladders, and it doesn't do windows. And, I don't mean the challenge to be one of developing a "Cyc-like" commonsense knowledge base and reasoning system. Our factotum will be allowed some lapses in commonsense so long as it can learn from its mistakes and benefit from instruction. It is not my purpose to set high standards for speech understanding and generation. The human task-givers should be tolerant of and adapt to the current limited abilities of systems to process natural language.

The second part of the challenge is that the robot must stay on-the-job and functioning for a year without being sent back to the factory for re-programming.

What will be required to meet this challenge? First, of course, a major project involving the application and extension of several robotic and AI technologies and architectures for integrating them. I do not think that it will be feasible for the robot's builders to send it to its office building with a suite of programs that anticipate *all* of the tasks that could be given. I think the robot will need to be able to *plan* and to *learn* how to perform some tasks that the building occupants (who know only about its sensors and effectors) might expect it to be able to perform but that its programmers did not happen to anticipate. To plan and to learn efficiently, I think it will need to be able to construct for itself hierarchies of useful action routines and the appropriate associated perceptual processing routines for guiding these actions. Perhaps something like the "twin-tower" architecture of James Albus (1991) would be appropriate for overall control. But the towers will have to grow with instruction and experience. The computational models of developmental learning proposed by Gary Drescher (1991) seem to me to be a good place to start for the tower-building aspect of the problem.

Work on this challenge problem would be good for AI. It would encourage progress on extending and integrating the many disparate components of intelligent systems: reacting, planning, learning, perception, and reasoning. It might also connect the bottom-up and top-down AI approaches — to the benefit of both. (It could also produce a useful factotum.) Good luck!

References

J.S. Albus. Outline for a Theory of Intelligence. *IEEE Systems, Man, and Cybernetics*, Vol 21, No. 3, pp. 473-509, May/June 1991.

G. Drescher. *Made Up Minds: A Constructivist Approach to Artificial Intelligence*, Cambridge, MA: MIT Press, 1991.

E.A. Feigenbaum, and J. Feldman. *Computers and Thought*, New York, NY: McGraw-Hill, 1963.

M. L. Ginsberg. Do computers need common sense? Techn. report, CIRL, Univerity of Oregon, 1996.

J. Hartmanis. Computational complexity of random access stored program machines. *Mathematical Systems Theory*, 5:232–245, 1971.

D.O. Hebb. *The Organization of Behavior.* John Wiley and Sons, New York, New York, 1949.

G. Kasparov. The day that I sensed a new kind of intelligence. *Time*, March 25, 1996, p. 55.

W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. of Math. Biophysics*, 5:115–137, 1943.

Marvin Minsky. *Computation: finite and infinite machines.* Prentice-Hall, 1967.

Marvin Minsky and Seymour Papert. *Perceptrons.* MIT Press, Cambridge, Massachusetts, 1969.

Allen Newell, J.C. Shaw, and Herbert Simon. Chess playing programs and the problem of complexity. *Journal of Research and Development*, 2:320–335, 1958. Also appeared in (Feigenbaum and Feldman 1963).

Oliver G. Selfridge. Pattern recognition and learning. In Colin Cherry, editor, *Proceedings of the Third London Symposium on Information Theory*, New York, New York, 1956. Academic Press.

Karl Sims. Evolving 3d morphology and behavior by competition. In Rodney A. Brooks and Pattie Maes, editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 28–39. MIT Press, Cambridge, Massachusetts, 1994.