

L. Stephen Coles, Alan M. Robb, Paul L. Sinclair, Michael H. Smith, and Ralph R. Sobek
 Department of Electrical Engineering and Computer Sciences
 University of California at Berkeley
 Berkeley, California 94720 U.S.A.

Abstract

This paper reports on the continuing design of and experimentation with Jason, the Berkeley Robot. Progress has been made in various aspects of the hardware (including the chassis, communications controller, and onboard microprocessor) and software (including problem-solving programs and world models). A particular experiment, analogous to the classical "Monkey and Bananas" Problem, is described. A major feature of the reformulation of this problem is the use of Decision Analysis in coping with uncertainty. Based on the accumulated expected costs of executing the steps of various hypothetical plans, Jason can evaluate the relative merits of direct action versus prior information-gathering using potentially unreliable sensors.

Introduction

As reported at the last IJCAI Conference,¹ the objective of our current research at the University of California at Berkeley is the design and implementation of a relatively inexpensive, general-purpose, computer-controlled, mobile robot. The Berkeley robot, dubbed Jason, was initially designed during the Spring of 1973 and tested off-line the following Summer. Hard-wire, blind-mode computer control was successfully accomplished during January 1974. Unreliable bread-board wiring as well as greater than expected demands for on-board power have led to a substantial redesign effort; testing of a considerably improved version of Jason is now underway.

Our ultimate goal, as stated earlier, remains the investigation of the class of problems that a robot both encounters and creates while performing elementary tasks in a real-world environment including active human beings. The results of this research will hopefully enable us to construct better and safer robots at a modest price that are still capable of performing a variety of useful tasks such as factory or ware-house work or can function as a teaching aid for young children in the classroom.

This paper is divided into five sections. The first briefly reviews some of the recent improvements to Jason hardware and software. The second section suggests some of the ways that robotics research differs from conventional AI problem-solving research. The third section presents a decision analysis formulation of the classical "monkey and bananas" problem. The fourth section provides the results of the decision analysis. Finally, the last section summarizes and proposes some improvements to Jason hardware and software for the future.

Jason Hardware/Software Improvements

Figures 1 and 2a-f provide an overview of the current state of Jason.

A. Hardware

First we discuss improvements to the key hardware subsystems: the chassis, the sensors, and the communications controller.

1. Robot Chassis--The Jason chassis consists of a half-inch thick aluminum base plate with an area of about two square feet with two additional aluminum shelves mounted above for electronic equipment. The three-ball configuration originally designed to support the front of Jason, although conceptually good, did not perform well on rough surfaces like a parking lot, and consequently was replaced by a heavy-duty, 3-inch diameter, swivel-caster wheel. To provide additional on-board power, the conventional, 83-amp-hour, lead storage, auto battery was replaced by a 250-amp-hour, 150-pound, train battery. Due to extensive two-handed coordination problems, the original plan for two simple hands has been reduced to one, and a prosthetic arm with jaw gripper has now been installed. A simple two-position-sensor push bar has also been installed: one setting indicates slight pressure contact while the other indicates too much pressure, i.e., Jason is trying to push a nonpushable object. The motor-control unit was also rebuilt to make it more rugged.

2. Sensors--An A-to-D converter to extract texture information as well as range data from the analog output of the ultrasonic torch has been designed. An LED proximity sensor has been mounted on the arm, and the proximity-sensor interface was completed.

3. Communications Controller--The preliminary Jason 8-bit character asynchronous communications controller was successfully bench tested, re-fabricated on circuit cards, and mounted in a new card rack and chassis by one of the authors (Robb). Previous problems of vibration should now be minimized, and circuit debugging should be greatly facilitated. Because the controller has been designed to be teletype-compatible, Jason could in principle be interfaced to any computer without special-purpose hardware being installed at the computer side. For example, Jason has already been connected to an HP-3000, a CDC-6400, and a PDP-10 computer system. Furthermore, we have recently demonstrated Jason over the ARPA Net. Further details on the controller can be found in Ref. 2. Our FCC radio-license was renewed, and two-way radio telemetry has been successfully bench tested.

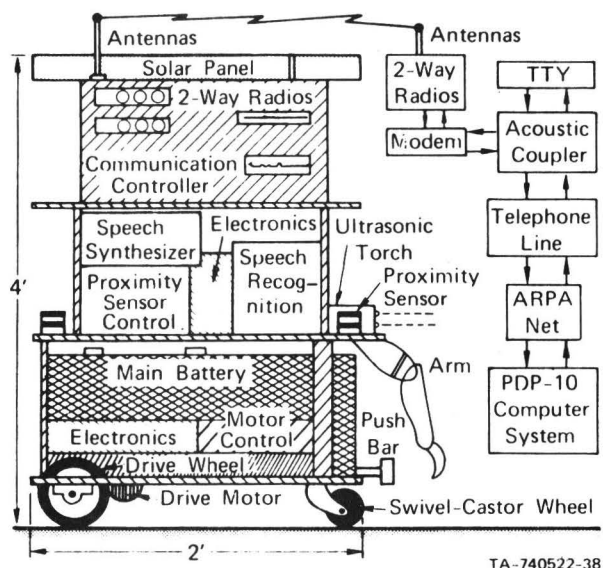


FIGURE 1 JASON, THE BERKELEY ROBOT

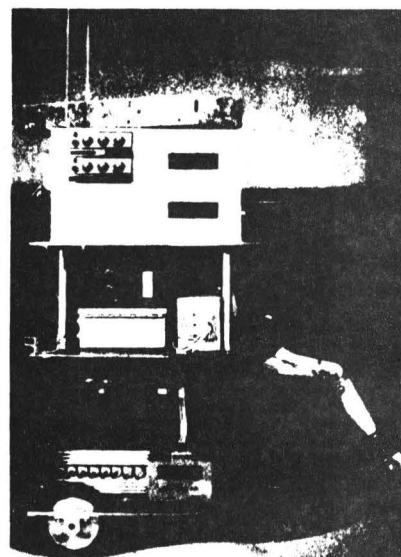


FIGURE 2a JASON-SIDE VIEW

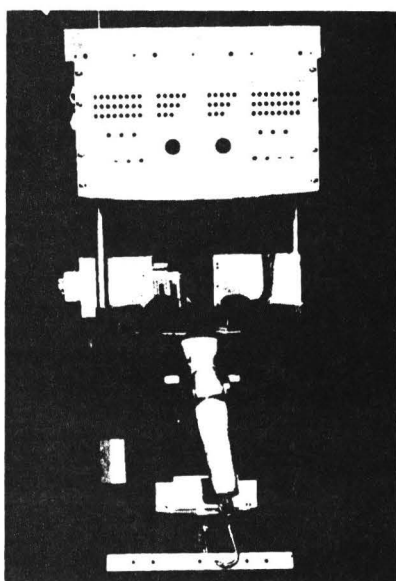


FIGURE 2b JASON-FRONT VIEW

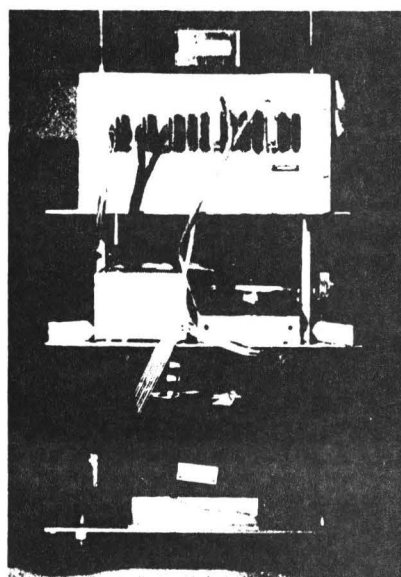


FIGURE 2c JASON-REAR VIEW

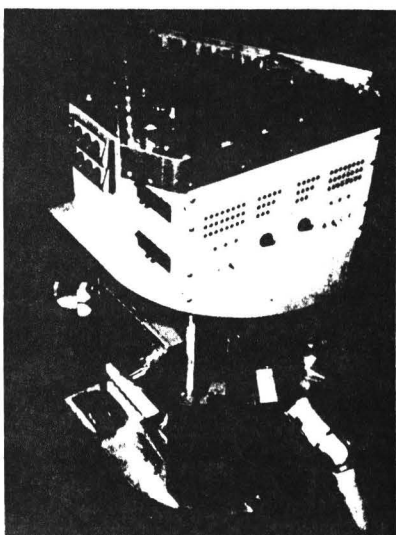


FIGURE 2d JASON-PERSPECTIVE VIEW

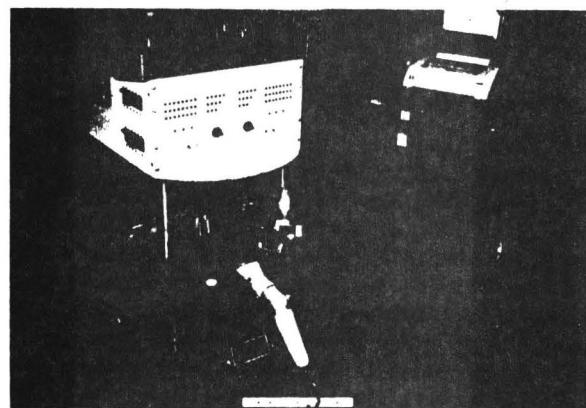


FIGURE 2e JASON WITH CONTROL TELETYPE

B. Software

On the software side, the check out of real-time interactive routines was handicapped by the removal of the HP-3000 computer from the Lawrence Hall of Science. Thus, all interactive work with Jason during the last year was simulated on the CDC-6400 available on the Berkeley campus. We have recently transferred Jason software to an interactive DEC PDP-10 computer system and have successfully tested Jason over the ARPA Net.

In the mean-time the Jason simulator, which has produced output tapes capable of actually driving Jason in blind mode, has grown extensively to include a much broader collection of Intermediate Level Operators (ILO's) such as Move, Turn, Face, Goto, Push, Pushto, Gotodoor, Gothrudoor, etc., capable of execution in an arbitrary collection of rooms connected by doorways and corridors, navigating optimally through an arbitrarily complex collection of boxes in any room. This work is documented in the Jason Reference Manual,³ and although it represents the work of many individuals, two of the authors (Sinclair and Sobek) are largely responsible. Output from the simulator will be seen in Section IV.

Real-World Vs. Toy Problems

Early research in artificial intelligence tended to focus on a few well-known puzzles or "toy" problems, such as the Monkey and Bananas, Missionaries and Cannibals, or Tower of Hanoi problems (see Ref. 4 for these and other examples). Although toy problems demonstrate the intellectual competence of computers along one narrow dimension, research in robotics has broadened considerably the scope of the problem-solving enterprise. Our experience with the Jason project permits us to identify at least four general criteria which distinguish real-world robot problems from toy problems: problem-solving environment, problem formulation, data requirements, and solution requirements.

A. Problem-Solving Environment

A rough spectrum of problem-solving environments in terms of increasing complexity is indicated in the following table:

	Environment	Description
E1.	Sterile	Mathematically precise, no interaction with the real world.
E2.	Surgically Clean	Real-world inputs, but highly contrived, such as painting objects with saturated colors so that the robot can identify them more easily.
E3.	Passive	Typical interior environment (such as an office) without people. Also, no wind or shadows due to changing illumination.
E4.	Active	Typical exterior environment (trees, sky, etc.), but without people or animals.
E5.	Benevolent	Active environment with cooperative humans exhibiting ruled behavior (such as in a factory or parking lot).
E6.	Natural	Benevolent environment including animals as well as seasonal weather changes.
E7.	Hostile	No constraints: mines, deep under water, outer space, etc., including potential adversaries.

TABLE 1 PROBLEM-SOLVING ENVIRONMENTS

In the language of Table 1, almost all work in AI was conducted in a sterile environment (E1). Subsequent early robotics research began with surgically-clean environments (E2) and was severely criticized for giving the appearance of E3 to uneducated observers, when in reality it was not very much of an advance over E1. As more advanced scene analysis and perceptual techniques became available, emphasis shifted to E3 and E4. With the appearance of more sophisticated hardware sensors initial forays are being made into benevolent environments (E5). All work to date in Natural or Hostile Environments (E6,7) has been done with teleoperators, where a human is an essential part of the loop.

B. Problem-Formulation

Because of the real-time execution and training aspect of robotics work, the problem formulation must be accomplished in human terms rather than mathematically. Thus, voice-input, restricted natural-language-problem statements are desired. Other techniques to facilitate man-machine interaction such as "joy-sticks," cursor tracking balls, or light pens are also needed.

C. Data Requirements

Information to solve the problem may either be inadequate or embedded in a large quantity of seductive but mostly superfluous data, or some combination of these. Insufficient information might occur for a variety of reasons: (i) it may not be knowable in principle, (ii) it may not be known whether it is knowable in principle, (iii) the necessary information is knowable, but the cost of acquiring it may be prohibitive. (As a special case, the cost may be reasonable, but acquisition cannot be accomplished within sufficient time to be useful), and (iv) it is not known whether the cost of information-gathering will prove to be prohibitive. (In this case decision analysis may be useful.)

Once obtained, data may still possess uncertainty or lack credibility for various reasons: (i) the data may be incorrect due to statistical unreliability in the sensory path, (ii) the source is known to be prejudiced, (iii) the source is

known to be antagonistic and may deliberately provide false leads. Finally, (iv) the data may be inconclusive because of inherent ambiguities within the model upon which it is based and may not yield a definitive interpretation. Medical or meteorological data fall in this category.

D. Solution Requirements

As distinguished from conventional AI problem solving, a solution is not a solution for a robot until it has been successfully implemented as action in the real world. There are many opportunities for failure along the path to a solution in this sense: planning failures, execution failures, and decision failures (a failure of communication between the planner and the executor).

1. Planner Failures--Within the planner various failure modes may occur: Either a plan is found or it isn't. If no plan is found, it could either be due to the fact that none exists (i.e., the task is truly impossible) or the system lacked the intelligence to find one. If a plan is found, it might still fail for a variety of reasons: (i) it never could have worked (i.e., it fails in principle); (ii) it sometimes works (i.e., it fails in practice). A plan is said to be incomplete⁵ (or soft) if it deliberately avoids dealing with all logically possible contingencies. Although such fragmentary plans are generally undesirable, they are sometimes preferable to no plan at all. Yet even a complete (or robust) plan will occasionally fail at some point in its capacity to sustain variation in environmental boundary conditions. If one were to plot the performance of a robot plan as a function of environmental complexity (E1 to E7 of Table 1) then one could characterize the failure mode as precipitous or capable of graceful degradation in proportion as whether the shape of performance fell off sharply or smoothly with complexity; (iii) even though questions of optimality are rarely stated explicitly in the problem formulation because it is normally unimportant whether the absolutely best way of doing something is proposed, plans should be penalized as failures if they are ludicrously inefficient in accomplishing the job. A plan post-processor might be useful in overcoming this kind of failure; (iv) the use of nondeterministic plans with parallel subsolution paths may make a plan more robust at the possible expense of introducing other failure modes.

2. Execution Failures--Execution failures come in two broad categories: internal and external. Internal failures sometimes called crashes, can either be hard or soft. A hard crash is not immediately recoverable, and the plan must be re-initialized. It may have been the result of either unreliable hardware or software. By comparison, a soft crash, usually due to a high-level monitor failure, will allow the robot to resume the plan where it left off, after a delay for reloading a fresh copy of the monitor. External failures are of three main types: (i) The robot failed, and knows that it failed. This may be due to either

systematic or random errors in its operators when executed in the real world, since all actions have inherent uncertainty in their outcomes. (Better calibration should hopefully minimize systematic errors.) There may also be other legal error modes for operators, such as "timeouts" or resource-exceeded constraints; (ii) the robot failed, and didn't know that it failed (sometimes called a missense error). This may be due to infinite looping or other errors in the flow-of-control or to self-deception through imperfect sensing of the true state of affairs; (iii) the robot succeeded, but thinks that it failed (sometimes called a nonense error). Illegal error messages or false alarms cause this kind of failure.

3. Decision Failures--Decision failures are much more subtle. Five general types will be mentioned: (i) too little or too much time devoted to planning compared with execution. This depends on the amount of time spent planning, the cost of thinking, the utility of the goal, the penalty for incurring undesirable irreversible state changes in the real world, and so forth. Note that human intuition may be very poor in this regard, since most human planning appears to take place at the subconscious level and therefore creates the illusion of being cost-free. Also, human planning and execution can frequently take place simultaneously, assuming that the execution process is not too intellectually-demanding. A robot may not always have this luxury; (ii) the failure to capitalize on serendipity. Tunnel vision during execution may cause the robot to push the solution out of the way in order to recreate the solution according to plan; (iii) failure to adequately reparameterize plans based on past experience. This is sometimes referred to as structural as distinguished from statistical learning; (iv) failure to reorder priorities dynamically. Multiple, possibly conflicting goals must be continually monitored during execution. This may lead to seemingly anomalous behavior from the point of view of an outside observer, but be perfectly consistent with internal objectives; (v) failure to distinguish local from global failures, i.e., calling upon the planner to replan from scratch, when salvaging the existing plan with a minor elaboration of an existing contingency branch of the current plan would be adequate, or conversely, trying all variations of a plan that was doomed to fail. In psychiatry this kind of pathology is referred to as functional fixity.

A New Formulation of the "Monkey and Bananas" Problem

The original formulation of the "Monkey and Bananas" problem⁶ can be stated briefly as follows:

A monkey is in a room in which a bunch of bananas are hanging from the ceiling, just out of reach. The monkey's problem, obviously, is to get the bananas. In the corner of the room is a chair. The solution

decided on by the monkey is to push the chair to a location under the bananas, climb on top of the chair, and then easily reach for the bananas.

The major interest of AI researchers in this problem is that it is characterized by one level of indirectness. That is, the solution requires an auxiliary device or tool (a chair in this case) not obviously needed at the start of the problem. In 1970 one of the authors (Coles) succeeded in formulating a fairly straight-forward transliteration of the problem into the world of the SRI robot.⁷ The role of the tool was played by a ramp that allowed Shakey to push a box off a platform onto the floor, which he would not otherwise have been able to do. Others, such as McDermott,⁸ have sought to generalize the information-gathering aspects of the problem.

In a more recent paper by Feldman and Sproull⁹ this same problem has served as the basis for a decision-theoretic approach. Although they are much more concerned with the use of a numerical utility function during planning to develop a more efficient search strategy, we have independently reached the same conclusion regarding the positive value of joining decision analysis with a symbolic robot problem solver to facilitate intelligent decision making under conditions of uncertainty. (See references 10-12 for an introduction to decision analysis.) In particular, using this approach we can create a decision tree that allows one to "roll back" the consequences of further information-gathering operations in comparison with direct action. Of course, decision analysis can never guarantee a desirable outcome; it can only provide reasons why one course of action will in general be better than another.

Because of the inherent unreliability of Jason's ultrasonic texture/range-finder, we sought to find a realistic reformulation of the monkey and bananas problem that would also illustrate the value of decision analysis when using this sensor. Figure 3 shows an initial plan view of Jason's world. It consists of two connecting rooms, R60 and R70, containing various boxes. Jason, currently in R60, is designated by the symbol "J" followed by an arrow indicating his principal orientation. The symbol "W" designates a wall, while the "+" sign indicates a clearance border for navigation purposes. Jason's only current goal is "IN(J,R70,100)," i.e., Jason desires to be in the adjoining room with a utility value of 100 ergs. Thus, the accumulated cost of all his effort (both planning and execution) in accomplishing this goal should not exceed 100 ergs; otherwise Jason will have wasted his energy.

Now in this formulation, although "thinking" is assumed to be free of charge, every operation Jason can carry out in the real world is energy consuming. Table 2 shows the approximate cost of an application of each of the ILOs relevant to this

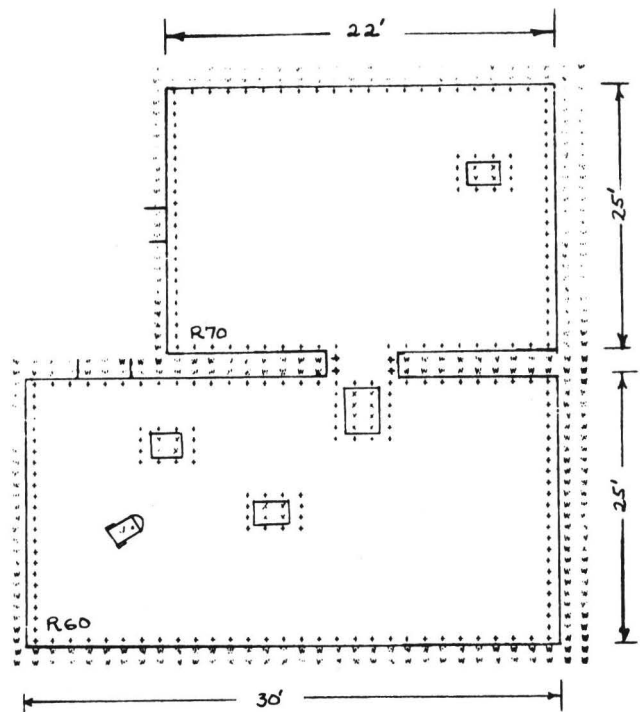


FIGURE 3 PLAN VIEW OF JASON'S WORLD problem.* Note that in general they entail a substantial start-up cost possibly plus an amount proportional to distance moved or turned. The proportionality constant, α , is on the order of 1/10 when distance, $d(x,y)$, is measured in feet. Also note that "pushing" is twice as expensive as "moving." The preconditions and postconditions indicated are typical of a STRIPS-like¹³ symbolic problem solver.

Under normal conditions the problem solver would yield the plan: (i) test for clear path to door D67 (connecting Rooms R60 and R70), (ii) Go to Door; (iii) Go through door. This could be accomplished for about 25 ergs, making it quite an attractive plan. However, in this formulation there is a box blocking the doorway, which would have caused "clearpath" to fail. Thus, the problem solver might instead generate the plan (i) Go to box; (ii) Push box (out of the way); (iii) Go to door; (iv) Go through door. This plan might be executed for about 35-40 ergs, still making it quite attractive.

However, in this particular world there is an additional complexity regarding the pushability of boxes. By definition, there are two general types of boxes: short and tall. Short boxes are not directly pushable because Jason's push bar would overhang them. Nevertheless, a short box could be pushed by means of a pushable tall box as shown in Figure 4. Furthermore, tall boxes come in two varieties: smooth and rough. Smooth boxes are

* Actual costs are accumulated by calls to Low Level Operators (LLOs) which may vary depending on the circumstances.

Operator	Description	Approximate Cost (ergs)	Preconditions	Postconditions		
				Delete List	Primary Add List	Secondary Add List
1. Clearpath(a)	Ultrasonic Torch determines whether an obstacle blocks a straight-line path to point a.	1	Face(J,a)		Clearpath(a)	
2. Texture(B,p)	Ultrasonic Torch determines whether the texture of box B is rough or smooth with probability p.	1	Face(J,B) IN(J,R $\leftarrow$ \$1) IN(B,R)		Texture(B)= Rough or Smooth	
3. Turn(t)	Jason turns t degrees.	$3 + \frac{\alpha}{45}t$	Clear(t) Angle(J, $\theta \leftarrow$ \$1)	Angle(J, θ)	Angle(J, $\theta + t$)	
4. Move(d)	Jason moves d feet.	$3 + \alpha d$	AT(J, a $\leftarrow$ \$1) Clearpath(a+d) IN(J, R $\leftarrow$ \$1) IN(J+d, R)	AT(J, a)	AT(J, a+d)	
5. Goto(a)	Jason goes to point a in the same room.	$5 + \alpha d(J,a)$	IN(J, R $\leftarrow$ \$1) IN(a, R)	AT(J, \$1)	AT(J, a)	
6. Gotodoor(D)	Jason goes to door D connecting the current room to another adjoining room.	$5 + \alpha d(J,D)$	IN(J, R $\leftarrow$ \$1) CONN(D, R, \$1)	AT(J, \$1)	AT(J, D)	
7. Gothrudoor(D)	Jason goes through door D.	5	AT(J, D) IN(J, R $\leftarrow$ \$1) CONN(D, R, R' $\leftarrow$ \$1)	IN(J, R)	IN(J, R')	
8. Push(B,a)	Jason pushes box B to point a.	$8 + 2\alpha d(B,a)$	Pushable(B) IN(J, R $\leftarrow$ \$1) IN(B, R) IN(a, R) AT(B, b $\leftarrow$ \$1) AT(J, b)	AT(B, b) AT(J, b)	AT(B, a)	AT(J, a)

Notes: $d(x,y)$ = distance from x to y. $\alpha \approx 1/10$. "\$1" matches a single arbitrary constituent. "x $\leftarrow$ \$1" means that x is bound to whatever value "\$1" matches.

TABLE 2 INTERMEDIATE LEVEL OPERATOR (ILO) DESCRIPTIONS

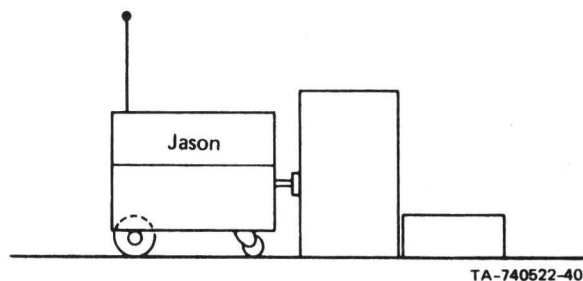


FIGURE 4 JASON USING A TALL BOX TO PUSH A SHORT BOX pushable, while rough boxes are nonpushable (because they're too heavy). Now if the box blocking

the doorway were tall, the preceding plan might be put into effect. However, it is known by Jason to be short. Thus, in such a case the existence of an available smooth tall box is a prerequisite to solving the problem. Now Room R60 contains not one, but two tall boxes. If Jason knew for a fact that at least one of the boxes were smooth and which one it was, the following plan might be generated: (i) Go to the smooth box; (ii) Push the smooth box to the low box (blocking the door); (iii) Push both boxes (out of the way); (iv) Go to door; (v) Go through door. Such a plan might be executed for about 60-70 ergs and corresponds closely to the original monkey and bananas problem. If both boxes were known to be smooth, Jason would choose to go to the box that minimized the combined cost of going and pushing. Note that the

assymmetric cost of these operations would in general place a premium on the box closer to the low box. Since both boxes are approximately equidistant to both Jason and the low box, Jason is indifferent according to this criterion.

However, in this case Jason does not know a priori whether neither, one, or both tall boxes are smooth. The a priori probability distribution of rough and smooth tall boxes is assumed to be uniform. Jason has available through its ultrasonic range-finder, a texture operation he can perform to determine whether a box is smooth with probability p , i.e.,

$\text{Prob}(\text{smooth}|\text{texture}=\text{smooth})=p$
 $\text{Prob}(\text{smooth}|\text{texture}=\text{rough})=1-p$
 $\text{Prob}(\text{rough}|\text{texture}=\text{smooth})=1-p$
 $\text{Prob}(\text{rough}|\text{texture}=\text{rough})=p$.

Using common sense, if p is very close to 1, then a cost of 1 erg to measure texture is likely to be a worthwhile investment compared to the risk of going to a possibly rough box while the other one might have been smooth. On the other hand as p approaches 0.5, the texture operation becomes increasingly unreliable and the marginal utility of the information obtained with respect to the cost of gathering it becomes smaller. Beyond some specified point of indifference as p approaches closer to 0.5, the cost of information gathering becomes prohibitive compared to its value. Testing texture would actually be counterproductive, and the best strategy would be to choose a box at random, go to it, and try to push it. In the worst case for which a solution were possible, i.e., the first box tried was rough while the other was smooth, the cost might be between 80 and 110 ergs, therefore still worth trying.

Now Jason has some empirical statistics on the reliability of his range finder that indicate that $p=0.95$. What should he do? Texture operations are expensive and risky. Yet any action may be fraught with peril. Jason turns to decision analysis to provide an answer. Figure 5 shows Jason's Decision Tree for the relevant portion of this problem. The results of the analysis are presented in the next section.

Results of the Decision Analysis

The internal representation of the decision tree for evaluation purposes is as a two-page "SEETREE" program¹⁴ which is very similar in appearance to Fortran IV. Each node, whether a decision node or a probability node, is defined as a separate function with its successor designated by the identifier "NEXT." The SEETREE program is translated into a standard FORTRAN IV program by a translator package operating over the ARPA Net at UCLA. Upon execution, the system generated a 56 node tree. Table 3 shows the "selected decision" for various values of p . The system also automatically provides rollback values of all subsequent decisions in the tree and the minimum, maximum, mean, and standard deviation of rewards for the

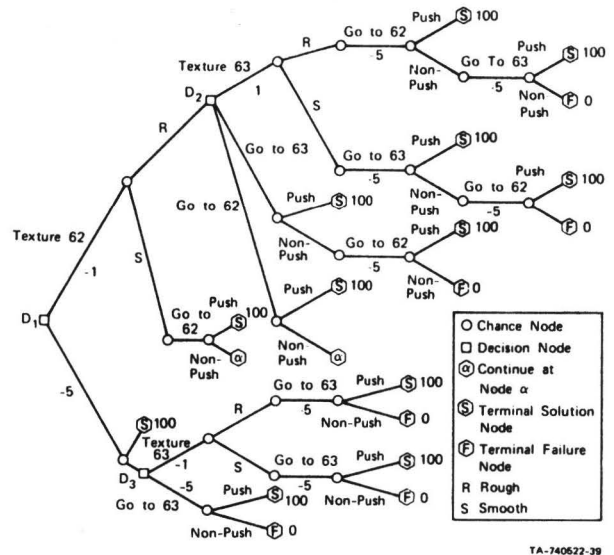


FIGURE 5 JASON'S DECISION TREE

lottery on the tree. A cumulative probability distribution of lottery values is also plotted automatically.

p	Selected Decision	Certain Equivalence	Certain Equivalence of Alternative Decision
0.80	GOTO Box62	67.500	67.250
0.90	GOTO Box62	67.500	67.500
0.95	TEXTURE Box62	67.625	67.500
1.00	TEXTURE Box62	67.750	67.500

TABLE 3 RESULTS OF DECISION TREE EXECUTION

Inspection of Table 3 reveals Jason's indifference point to be $p=0.9$, since the certain equivalence for either decision is the same. Since Jason's historical data suggests a value of $p=0.95$, his optimal strategy should be to measure texture before going to either box. If the sensor reports back "smooth," then he should go immediately to that box. Otherwise he should go directly to the other box without further measurement, since further increase in the confidence of its true texture would not make a difference in Jason's subsequent behavior.*

* Providing $p < 1$. Of course, if $p=1$ and the first box was discovered to be rough, then testing the texture of the second box would be useful: if it too were rough, Jason could definitively give up without moving. However, if there is even the slightest possibility that the sensor was in error ($p < 1$), Jason would be compelled, lacking other alternatives, to verify a box's roughness empirically by going to it and trying to push it.

756

Figure 7 shows the form of the plan output by the symbolic problem solver (comments inserted between slashes). Plan1 is then executed interpretively. In this case under the assumption that the texture of Box62 is reported to be smooth and further that it is actually smooth, and therefore pushable, the final cost of the plan under simulated execution is 92.95 ergs. Since this value is within the specified utility of 100, Jason has achieved a net profit of over 7 ergs.

Figure 8 shows the initial state of R60 with the proposed trajectory marked by arrows ("+" signs have been suppressed.). In Figure 10 Jason has pushed both B61 and B62 out of the way. Finally, Figure 11 shows Jason having executed the Gotodoor/Gothrudoor portion of his plan. In execution, this plan produced a string of 19 individual move and turn commands, each of which generates its own motor command or Low Level Operator (LLO), resulting in a 50-page listing of grid positions. The simulated execution took approximately 3 seconds on the CDC-6400 and 7.6 seconds on the PDP-10. Based on prior experience, we expect that actual execution in the real world will take on the order of 3 minutes.

Summary and Future Work

The major contribution of this paper is a demonstration, by means of a well known example, of how decision analysis can be used to improve the decision-making capability of a robot under conditions of uncertainty in its perceptual inputs. Now that Jason has been demonstrated over the ARPA Net, our next priority will be to link the decision analysis software with the Jason control software as an integrated system. We can then experiment dynamically with different boundary conditions as well as different levels of reliability for texture measurements.

In the longer term we have outlined a number of objectives. Our intermediate goals include the development of software to handle (i) multiple (possibly conflicting) time-dependent goals, (ii) dynamic tracking (of a cooperative agent), and (iii) dynamic real-time collision avoidance during navigation, including the evasion of active agents such as slowly moving people in a crowded corridor. Our long term goals include the application of a Jason-like robot to factory or warehouse work where the environment is fairly well controlled.

The Berkeley robot project is still in the midst of design and construction. Many devices have been designed, tested, and installed. A considerable amount of software has been written and debugged. Other devices and programs are still in the development stage. It is planned that a fully-integrated robot vehicle will again be operational in the near future. Once operational, Jason will be used as a test bed for the development of future, general-purpose mobile robots with even greater sophistication and intelligence.

Acknowledgments

The authors would like to thank the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley for its continuing support of the Jason Project. In particular, Prof. David Hodges and Mr. Leonard Baldwin of the E.E.C.S. machine shop deserve special mention. We also wish to acknowledge the assistance of Dr. Ramon Zamora of the Decision Analysis Group at Stanford Research Institute.

Bibliography

1. Michael H. Smith and L. Stephen Coles, "Design of a Low-Cost General-Purpose Robot," Proc. of the IJCAI-73, pp. 324-335 (August 1973).
2. Alan Robb, "A Design of a Communications Controller for Jason, the Berkeley Robot," EECS 292H Project Report, The University of California at Berkeley (June 1974).
3. Paul L. Sinclair and Ralph P. Sobek, "Jason Reference Manual," EECS 292H Project Report, The University of California at Berkeley (June 1974).
4. George W. Ernst and Allen Newell, GPS: A Case Study in Generality and Problem Solving (Academic Press, 1967).
5. John H. Munson, "Robot Planning, Execution, and Monitoring in an Uncertain Environment," Proc. of the IJCAI-71, pp. 338-349 (September 1971).
6. John McCarthy, "Situations, Actions, and Causal Laws," Memo No. 2, Stanford University Artificial Intelligence Project, Stanford, California (July 1963).
7. L. Stephen Coles, "An Experiment in Robot Tool Using," Proc. of the IEEE Systems, Man, and Cybernetics Society International Conference (Pittsburgh, Pennsylvania, 1970).
8. Drew V. McDermott, "Assimilation of New Information by a Natural Language-Understanding System," M.S. Thesis, TR-291, MIT Artificial Intelligence Laboratory (February 1974).
9. Jerome A. Feldman and Robert A. Sproull, "Decision Theory and Artificial Intelligence II: The Hungry Monkey," Technical Report No. 2, Computer Science Department, University of Rochester, New York (November 1974).
10. Warner North, "A Tutorial Introduction to Decision Theory," IEEE Transactions on Systems Sciences and Cybernetics, Vol. SSC-4, No. 3, pp. 200-210 (September 1968).
11. Ronald A. Howard, "The Foundations of Decision Analysis," IEEE Transactions on Systems Sciences and Cybernetics, Vol. SSC-4, No. 3, pp. 211-219 (September 1968).
12. Herman Chernoff and Lincoln E. Moses, Elementary Decision Theory (John Wiley and Sons, New York, 1959).
13. Richard E. Fikes and Nils J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving," Artificial Intelligence, Vol. 2, Nos. 3/4, pp. 189-208 (Winter 1971).
14. Ramon M. Zamora and Ellen B. Leaf, "Tutorial on the Use of the SRI TREE Language System" Technical Memorandum, Stanford Research Institute, Menlo Park, California (December 1974).