An Analysis of Error Recovery and Sensory Integration for Dynamic Planners *

Bruce Abramson
Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0782
bda@cse.usc.edu

Abstract

Strategic planners for robots designed to operate in a dynamic environment must be able to decide (i) how often a sensory request should be granted, and (ii) how to recover from a detected error. This paper derives closed-form formulas for the appropriate frequency of sensor integration as a function of parameters of the equipment, the domain, and the types of errors from which the system wishes to recover.

Introduction

All behavior is guided by beliefs. People devise plans based on their perceptions of the world and then attempt to implement those plans. Robots intended to interact with the physical world are no different; they too must base plans on their internal models and then execute their plans in the physical world. Thus, strategic planners for these robots must coordinate sensory acquisition with robotic activity.

Classical planners focus only on activity; plans are translated directly into actions, with no provisions made for errors or spontaneous changes. Planners capable of sensing their environments, updating their plans, and acting according to changing beliefs may be termed dynamic planners. Any such planning paradigm must recognize the difference between its own internal model (beliefs) and the external world (facts). Sensors perform the crucial task of incorporating newly acquired information into a belief system.

This paper casts sensory coordination as a *decision* problem; it assumes that although relevant sensory data can generally be requested, costs prohibit many of them from being granted. Thus, the underlying decision problem is whether a central controller should grant or deny a system's request for a sensory update. Neither this problem nor this approach are unique. In fact, our work is similar in spirit to that of at least three groups of AI researchers: those interested in the applicability of general probabilistic, statistical, and decision theoretic concepts to AI [Cheeseman, 1988] [Horvitz *et al.*, 1988] [Ng and Abramson, 1990], those who have applied these concepts to simplified games and toys [Russell and Wefald, 1989] [Abramson, 1990a] [Hansson and Mayer, 1990], and those concerned with planning under uncertainty [Wellman, 1990] [Maes and Brooks, 1990]. The most directly relevant work, however, is in [Abramson, 1990b] and [Zerroug and Abramson, 1990], where we forwarded *skepticism* as the basis of a four-layer architecture for dynamic planning through a probabilistic state-space. The top layer of a skeptical architecture contains a STRIPS-like *planner* generates goal-directed plans. One layer down, a *skeptic* attempts to verify the planner's beliefs, by communicating with the (third layer's) *interpreters,* which, in turn, transmit requests to and sensory information from the (fourth layer's) sensors.

Our skeptical architecture's planning paradigm [Zerroug and Abramson, 1990] combined ideas from four different sources to suggest an approach towards sensory coordination: (i) a classical planning system, STRIPS [Fikes and Nilsson, 1971], (ii) the distinctions between standard controlled state-spaces (CSS's), in which everything occurs as planned, and probabilistic state-spaces (PSS's), in which nature occasionally causes unexpected occurrences [Abramson, 1990b], (iii) Georgeff's 1987 proposal that the frame problem may be solvable by allowing all items unaffected by an operator to vary freely, and (iv) the JPL Telerobot's automatic insertion of relevant sensory requests into a STRIPS-like plan [Doyle *et al.*, 1986]. These ideas are assumed throughout the paper; they help motivate and justify many of our underlying assumptions.

Although the results presented in this paper are offshoots of that research, (and a skeptical architecture is assumed throughout), the architecture's details are of purely tangential relevance. The key similarity lies in their common philosophical underpinning: Strategic planners must coordinate thinking (devising a PLAN), executing (performing an ACT), and SENSEing. All PLANs are devised on internal models, or belief systems. All ACTs are performed in the physical world. SENSE operations help reconcile internal beliefs with

*This research was supported in part by the National Science Foundation under grant IRI-8910173.

external reality. Thus, the first fundamental difficulty with which dynamic planners must cope is the strategic coordination of SENSEs, ACTs, and PLANs. This paper sets aside implementation issues in favor of analysis. Given our characterization of sensors as mechanisms that reconcile belief and reality, two important questions arise:

- How can a system recover from a detected discrepancy between belief and reality?
- How often must sensors be used? If sensing is relatively expensive, can resources be conserved by sensing only occasionally?

The basic setting underlying the following discussion, then, is as follows: A robotic system is given a task characterized as a search space with initial state $\mathcal{S}_0$ and goal state $\mathcal{G}$. It devises a plan which—if entirely successful—should solve the problem. At some point, however, it doubts (as a skeptic) that its existing state in the physical world, $\mathcal{S}_e$, is identical to the state in which it believes itself to be, $\mathcal{S}_b$. Should it grant a sensory request to determine its true status? If it grants the request and finds itself lost, how should it recover?

One of the greatest difficulties underlying error detection and recovery is the determination of reasonable assumptions about an an external environment that has deviated from its predicted course. For this reason, the model construction phase must focus on encoding distributions of likely failures for each action and reasonable consequences of "nature." Successful error recovery depends, to a great extent, on how much is known about the deviation of the internal model from external reality, and on the impact of this discrepancy on the existing plan. Encoded failure states and natural occurrences are designed to provide maximal information. Knowledge of equipment reliability and the effects of nature, however, are simply prerequisites to the design of a functioning system; encoded information alone is insufficient. Thus, our skeptical architecture contained many areas in which specific and carefully encoded information could later be added, and our analysis relates sensor integration to various parameters of the equipment, the environment, and the type of errors from which the system expects to recover.

Strategy Selection

Realistic error recovery schemes are hard to define and harder to analyze. Perhaps the simplest possible approach to "recovery" is to discard (automatically and immediately) all pending actions and to replan from the existing state, $\mathcal{S}_e$, to the goal, $\mathcal{G}$. This replan-from-scratch (SCRATCH) approach may become prohibitively expensive if spontaneous changes and/or errors occur frequently, and is unnecessary if the detected discrepancies only affect a small portion of the plan. A second strategy is to replan locally (LOCAL)—from $\mathcal{S}_e$ back to $\mathcal{S}_b$—to return to the state believed by the system, and then to proceed as intended. Initial implementations of the skeptical architecture adopted LOCAL [Zerroug and Abramson, 1990].

Analyses of SCRATCH, LOCAL, and other error recovery schemes will require quite a few simplifying assumptions and a good deal of notation. Terminology will be introduced as needed. As a first simplification, assume that the problem facing the skeptical planner is the standard one-player task: find the shortest path through a PSS, from $\mathcal{S}_0$ to $\mathcal{G}$ (PLAN). The goal of the skeptical robot is then to execute the plan (ACT). A PSS, like all state-spaces, can be described as a search graph of average branching factor B, and a distance (or optimal path length) of D_0 between $\mathcal{S}_0$ and $\mathcal{G}$. Let each planning step cost P and each execution step cost E.

To summarize the notation, then:

$\mathcal{S}_0$: initial state

$\mathcal{G}$: goal state

B: average branching factor of the PSS

D_0: distance (length of optimal path) from $\mathcal{S}_0$ to $\mathcal{G}$

P: cost of a single planning step

E: cost of a single execution step

A planner with access to no special heuristics must devise its plans by an $O(B^{D_0})$ brute force (generally breadth first) search, at a cost of $(P * B^{D_0})$. (The purpose of most existing planning paradigms, of course, is to avoid the combinatorial explosion inherent in brute force searches. This assumption is made to ease analysis, not to reflect reality). A perfect, error-free execution will then take D_0 steps, for an additional cost of $(E * D_0)$. The total cost for an ideal CSS-type plan and execution, with no error detection or correction mechanism and no sensory capabilities, forms a lower bound, (Λ), on the cost of a PSS-based paradigm.

$$C[IDEAL] = PB^{D_0} + ED_0 \equiv \Lambda \tag{1}$$

The advantage of the PSS assumptions, however, is that they allow nature to romp through the space and cause "spontaneous" changes. Let D_1 represent the number of operators from the original plan that have already been successfully applied, and let D_2 represent the number that remain, $(D_0 = D_1 + D_2)$. One of the skeptic's queries to the interpreters suddenly reveals that it is not in state $\mathcal{S}_b$, as it believed, but rather ϵ steps away, in state $\mathcal{S}_e$.

Additional notation now includes:

Λ: theoretical lower cost bound of a successfully devised and executed executed PLAN

C: a cost function

$\mathcal{S}_b$: state that the system believes exists

$\mathcal{S}_e$: state that exists in the physical world

D_1: number of successfully applied operators from the original plan

D_2: number of remaining operators from the original plan

ϵ: distance between $\mathcal{S}_b$ and $\mathcal{S}_e$

The system, of course, has already incurred the cost of devising the initial plan and executing D_1 of its steps, for an expenditure to date of $(P*B^{D_0}+E*D_1)$. It must now recover before proceeding. Since the first error recovery mechanism under consideration is SCRATCH, the planner must replan from $\mathcal{S}_e$ to $\mathcal{G}$. What is the distance between these two states? The worst case, of course, occurs when the error threw the system ϵ steps backwards (i.e., towards $\mathcal{S}_0$), making the $\mathcal{S}_e$-to-$\mathcal{G}$ distance $(D_2 + \epsilon)$. The best case, on the other hand, arises when the spontaneous change worked towards $\mathcal{G}$, reducing the distance to $(D_2 - \epsilon)$. Since the purpose of this analysis is to show the superiority of LOCAL to SCRATCH, the deck will be stacked against the favorite; spontaneous moves towards the goal save time for SCRATCH, but cause redundancies in LOCAL, and will thus be assumed. Given this scenario, then, SCRATCH's replan will cost $(P * B^{D_2-\epsilon})$ and its completion of the execution will cost $(E * (D_2 - \epsilon))$. Thus, the total cost of planning, starting execution, noting an error, replanning from scratch, and completing execution, is:

$$C[SCRATCH] = PB^{D_0} + ED_1 + PB^{D_2-\epsilon} + E(D_2 - \epsilon)$$
$$= \Lambda + PB^{D_2-\epsilon} - E\epsilon \qquad (2)$$

Unlike SCRATCH, LOCAL's replanning cost is insensitive to the direction in which $\mathcal{S}_e$ lies; it always replans and executes through $\mathcal{S}_b$, even at the expense of redundant operators. Thus, under any set of circumstances LOCAL's replan costs $(P * B^{\epsilon})$ and its completion of the execution costs $(E * (D_2 + \epsilon))$, or:

$$C[LOCAL] = PB^{D_0} + ED_1 + PB^{\epsilon} + E(D_2 + \epsilon)$$
$$= \Lambda + PB^{\epsilon} + E\epsilon \qquad (3)$$

LOCAL is preferable to SCRATCH, then, whenever it is the lower-cost option, or whenever $C[LOCAL] < C[SCRATCH]$. Equations 2 and 3 thus combine to recommend LOCAL over SCRATCH whenever:

$$PB^{\epsilon} + E\epsilon < PB^{D_2-\epsilon} - E\epsilon \Longrightarrow$$
$$2E\epsilon < P(B^{D_2-\epsilon} - B^{\epsilon}) \qquad (4)$$

It it not too difficult to see that even under the realistic assumption that execution is much more expensive than planning, $(E \gg P)$, SCRATCH is only preferable to LOCAL if either D_2 is quite small, ϵ is quite large, or both. In all other (i.e., nearly all) instances, LOCAL is the better replanning strategy. Heuristic information, of course, could improve the efficiency of both strategies, as well as avoid some of the redundancies inherent in LOCAL's approach. Nevertheless, heuristics equally applicable to both approaches should not affect their relative strength; LOCAL should emerge as general-case superior.

Sensory Coordination

Given the preference for LOCAL over SCRATCH implied above, the next question is what to do about error detection. When should a sensory request be granted (and downloaded to the sensors), and when should it be denied (and the belief acted upon as if true)? Before answering, a few more bits of notation are required. Let S refer to the cost of a single sensory operation. (In general, it is probably safe to assume $P < S < E$). Verification of a belief requires only one operation, so granting a sensory request that verifies the system's beliefs incurs a total cost of S. Determining the system's precise state when it discovers that it is not where it thought it was is much harder. Sensing to find $\mathcal{S}_e$ from $\mathcal{S}_b$, in fact, requires the same brute-force approach as did planning between them, for a cost of $(S * B^{\epsilon})$.

The frequency with which sensory operations are really needed, in turn, depends on the probability with which spontaneous changes occur. (In a realistic setting, this probability must be assessed from equipment and domain experts). Let p represent the probability with which a single unit of time will pass without a spontaneous change occurring. Let t represent the length of time needed to execute D_1 moves. Assume that at most one spontaneous change will occur during this time period, and that it is equally likely to occur during any given unit of time. Then the probability with which no change occurred while the first D_1 operators were being applied is p^t, and the probability with which a change did occur is $(1 - p^t)$.

Next, consider the consequences of denying a sensory request when one is needed. If the request should have been granted, the system must already be ϵ away from where it thought it was. By failing to grant the request, the system continues to apply operators and drifts even further off course. By the time that a later sensory request is granted, then, the system has drifted an additional δ steps away, to $\mathcal{S}_d$; it must now retrace the $(\delta + \epsilon)$ steps (in the worst case) back to $\mathcal{S}_b$ before the original plan may be resumed.

A sensory request should be granted whenever the expected cost of granting it, $(EC[G])$, is less than the expected cost of denying it, $(EC[D])$. In order to calculate these expected costs, four different cases must be considered: (i) an unnecessary request was denied (DU), (ii) a necessary request was denied (DN), (iii) an unnecessary request was granted (GU), and (iv) a necessary request was granted (GN).

One further summary of notation:

$\mathcal{S}_d$: state reached from $\mathcal{S}_e$ after a necessary sensory request was rejected

δ: distance between $\mathcal{S}_e$ and $\mathcal{S}_d$

S: cost of a single sensory operation

t: time needed to apply D_1 operators

p: probability of no spontaneous change in a single time unit

- $(1-p)$: probability of a spontaneous change occurring in a single time unit
- p^t: probability of no spontaneous change in time t
- $(1-p^t)$: probability that a spontaneous change will occur in time t
- EC: an expected cost function (expected value of C)
- G: decision to grant a sensory request
- D: decision to deny a sensory request
- N: instance in which a SENSE is needed
- U: instance in which no SENSE is needed

First, consider DU. If no sensing was needed, no change occurred. If none was granted, no expense was incurred. Thus, DU describes the CSS ideal and hits the lower bound of cost:

$$C[DU] = \Lambda \tag{5}$$

Second, consider DN. An error occurred and a sensory operation was needed, but the request was denied. The system executes another δ operators and drifts further away. A later (granted) SENSE detects the error and uses LOCAL to correct it. The cost for DN, then, is:

$$\begin{aligned} C[DN] &= \\ PB^{D_0} + ED_1 + E\delta &+ (S+P)B^{\delta+\epsilon} + E(D_2+\delta+\epsilon) \\ &= \Lambda + (S+P)B^{\delta+\epsilon} + E(2\delta+\epsilon) \end{aligned} \tag{6}$$

Third, consider GU. Planning and execution proceed as in the ideal; the only added expense is the single sensory operation used to verify some belief. Thus,

$$C[GU] = \Lambda + S \tag{7}$$

Fourth, consider GN. Some change occurred, but it was detected by a (granted) SENSE and corrected by LOCAL. Thus, DN's cost is given by:

$$\begin{aligned} C[GN] &= PB^{D_0} + ED_1 + SB^{\epsilon} + PB^{\epsilon} + E(D_2+\epsilon) \\ &= \Lambda + (S+P)B^{\epsilon} + E\epsilon \end{aligned} \tag{8}$$

The cost functions of equations 5, 6, 7, and 8, can now be combined to calculate $EC[G]$ and $EC[D]$. In both cases, expected cost is determined by considering the probability with which a request was needed, $(1-p^t)$, vs. the probability with which it was unnecessary, (p^t), and the costs incurred with or without that need. Thus:

$$\begin{aligned} EC[D] &= (1-p^t)C[DN] + p^tC[DU] \\ &= (1-p^t)(\Lambda + (S+P)B^{\delta+\epsilon} + E(2\delta+\epsilon)) + \Lambda p^t \\ &= \Lambda + (1-p^t)((S+P)B^{\delta+\epsilon} + E(2\delta+\epsilon)) \end{aligned} \tag{9}$$

$$\begin{aligned} EC[G] &= (1-p^t)C[GN] + p^tC[GU] \\ &= (1-p^t)(\Lambda + (S+P)B^{\epsilon} + E\epsilon) + p^t(\Lambda + S) \\ &= \Lambda + (1-p^t)((S+P)B^{\epsilon} + E\epsilon) + p^t(S) \end{aligned} \tag{10}$$

Equations 9 and 10, taken together, indicate whether a sensory request should be granted or denied. In general, a request should be granted when $EC[G] < EC[D]$, denied otherwise.

$$\begin{aligned} & EC[G] < EC[D] \Longrightarrow \\ & (1-p^t)((S+P)B^{\epsilon} + E\epsilon) + p^t(S) < \\ & (1-p^t)((S+P)B^{\delta+\epsilon} + E(2\delta+\epsilon)) \\ & \Longrightarrow p^t(S) < \\ & (1-p^t)((S+P)B^{\delta+\epsilon} + E(2\delta+\epsilon)) - \\ & (1-p^t)((S+P)B^{\epsilon} + E\epsilon) \\ & \Longrightarrow p^t(S) < \\ & (1-p^t)((S+P)(B^{\delta+\epsilon} - B^{\epsilon}) + E(2\delta)) \Longrightarrow \end{aligned}$$

$$\frac{p^t}{(1-p^t)} < \frac{(S+P)(B^{\delta+\epsilon} - B^{\epsilon}) + 2E\delta}{S} \tag{11}$$

– or –

$$\frac{(1-p^t)}{p^t} > \frac{S}{(S+P)(B^{\delta+\epsilon} - B^{\epsilon}) + 2E\delta} \tag{12}$$

The left-hand sides of equations 11 and 12 represent the *odds* of p^t and $(1-p^t)$, respectively, or the odds that a spontaneous change will occur during the application of the original plan's first D_1 operators. Although odds are a fairly basic statistical concept, their mention often suggests the terminology of the casino. In a very real sense, the decision to grant or deny a sensory request is a gamble. An extremely conservative (or completely risk averse) player would grant every sensory request despite the cost; this strategy is quite expensive, but it does guarantee the immediate detection of all deviations from the plan. An extremely liberal gambler (or a complete risk taker) would reject every request; cost is minimized, but the probability of successfully implementing the plan drops to p^{t_0}, where t_0 is the number of time units necessary to apply all D_0 operators. A rational player balances these concerns; he or she accepts reasonable risks, rejects unreasonable ones. A rational skeptic (the layer of the [Zerroug and Abramson, 1990] architecture charged with granting or denying sensory requests), must accept only reasonable risks when evaluating the merit of a sensory request.

Equation 12 defines reasonable risk. When the odds of a spontaneous occurrence surpass the threshold given by equation 12's right hand side, the stakes are too great for the skeptic; the request is granted. When they are below the threshold, on the other hand, a gamble (in fact, a parlay) is advised; the request is denied. Even given this sage advice, however, skeptical planners, like most gamblers, can't really win; the best that they can do is minimize their losses. The best (and lowest cost) case for the planner is DU, where it breaks even (nothing ventured, nothing lost). In all other cases, it loses— less in GU than in GN and less in GN than in DN—but it loses all the same. The story, however, is not hopeless. This entire section has been devoted to a study of costs. The existence of benefits is a hidden but ubiquitous assumption. A truly rational (in the decision theoretic sense) skeptical planner, then, attempts to solve all problems (i.e., find all paths from $\mathcal{S}_0$ to $\mathcal{G}$) whose expected benefits outweigh its expected costs; the combination of equation 12 and LOCAL offers an error detection and correction strategy designed to reach $\mathcal{G}$ while minimizing expected costs.

Frequency Analysis

The formulas derived in the previous section cast the decision to grant or deny a sensory request as a gamble; odds were devised to make the gamble fair. There are, however, many other ways to express the same idea. Since one of the variables included in equations 11 and 12, is length of time between senses, it should be possible to solve for t. The frequency with which sensory requests should be granted, then, is given by:

$$t < log_p\left(\frac{\frac{(S+P)(B^{\delta+\epsilon}-B^{\epsilon})+2E\delta}{S}}{1+\frac{(S+P)(B^{\delta+\epsilon}-B^{\epsilon})+2E\delta}{S}}\right) \quad (13)$$

The frequency of SENSEs is more than just an indication of how often sensory equipment should be used; the entire purpose of a SENSE is to update the system's internal models. Thus, the suggested time lapse between SENSEs also indicates the suggested lifetime of an internal model. Decisions based on models that have not been refreshed for longer than the advised time period are too risky to be made with comfort; insistence upon ACTing on old (i.e., past the expiration date) beliefs is a sign of irrationality. Equation 13, in turn, addresses the proper frequency of sensing. Experimentation with a few scenarios—assignments of values to S, P, E, ϵ, δ, and B—can easily demonstrate the importance of strategically coordinated sensing. (Throughout the rest of this discussion, assume that time is discretized into seconds; all values of t should be read as the number of seconds between granted sensory requests).

Scenario 1 The first scenario approximates the CSS ideal without actually achieving it. Equipment better than state-of-the-art makes planning, sensing, and acting equally inexpensive ($P = S = E$), and spontaneous changes never move the system more than one state away from where it should be ($\epsilon = \delta = 1$). The PSS is quite simple—there are, on the average, only three options open to the system at any given moment ($B = 3$). An almost perfectly controlled environment results in less than one spontaneous occurrence per day (86,400 seconds). Thus, $(1-p) = 1/86400 \approx 1.16 * 10^{-5}$, and p may be set to 0.99999. Plugging these numbers into right hand side of equation 11 yields:

$$\frac{(S+P)(B^{\delta+\epsilon}-B^{\epsilon})+2E\delta}{S} = \frac{(2P)(3^2-3)+2P}{P} = 14$$

$$\frac{p^t}{(1-p^t)} < 14 \Longrightarrow p^t < 14/15 \Longrightarrow (.99999)^t < .933333$$
$$\Longrightarrow t \approx 6900$$

In words, then, even a planning/acting system that interacts with a near-ideal environment needs to SENSE its surroundings occasionally. In this scenario, sensory requests should be granted about every 6900 seconds, or slightly more often than once every two hours.

Scenario 2 Most of the assumptions of scenario 1 still hold. The only change comes in the behavior of nature. Rather than occurring about once a day, spontaneous changes happen about once an hour (3600 seconds). Thus, $(1-p) = 1/3600 \approx 2.78 * 10^{-4}$, and p may be set to 0.9997. This change to p means that:

$$(.9997)^t < .933333 \Longrightarrow t \approx 230$$

Once-an-hour occurrences, then, recommend granting a sensory request about every four minutes.

Scenario 3 Nature is as well-behaved as it was in scenario 1, but the entire domain is much more complex. Thus, although p can be set once more to 0.99999, the system finds itself faced with many available options (say $B = 10$), and spontaneous changes that shift it two moves away from its believed state ($\epsilon = \delta = 2$). Nor is the equipment as advanced as it was in scenarios 1, 2, and 3; sensing is ten times as costly as planning, and acting twice as expensive as sensing ($S = 10P$, $E = 2S = 20P$). Given these values, equation 11 yields:

$$\frac{(S+P)(B^{\delta+\epsilon}-B^{\epsilon})+2E\delta}{S}$$
$$= \frac{(11P)(10^4-10^2)+80P}{10P} = 10898$$
$$\frac{p^t}{(1-p^t)} < 10898 \Longrightarrow p^t < 10898/10899$$
$$\Longrightarrow (.99999)^t < .99991 \Longrightarrow t \approx 10$$

Even an almost perfectly cooperative nature, then, can not save a planning/execution system from having to sense this environment fairly regularly. Despite the rarity of spontaneous change, the number of things that could go wrong (recall $B = 10$) and the cost of error recovery combine to dictate a SENSE about once every ten seconds.

Scenario 4 The assumptions of scenario 3 remain, but spontaneous changes happen about once an hour (3600 seconds), as in scenario 2. Thus,

$$(.9997)^t < .99991 \Longrightarrow t < 1$$

Once-an-hour occurrences in an even marginally complicated domain, then, argue in favor of almost continuous sensing.

Discussion

The scenarios discussed above illustrate, in a rather formal manner, why classical planners failed. It is difficult to imagine a more idyllic setting than the one described in scenario 1 (other than, perhaps, pure simulation mode). Even there, however, sensory requests should have been granted every two hours; as nature became less cooperative (scenario 2), the recommended time lapse between granted requests declined. The setting described in scenario 3 was more complex, but not particularly so; even a wonderfully cooperative nature needed to be investigated frequently. Scenario 4 mandated near-continuous sensing. The impact of the various parameters on sensory coordination gives some indication to the propriety of various types of planning paradigms in different environments. A classical CSS-based planner will never work (for very long) outside of simulation; it needs the ability to detect and correct errors. (This observation is not subtle. The scenarios simply formalize an intuition that has long been widely held). A Telerobot-type decision to grant all sensory requests [Doyle *et al.*, 1986] probably represents overkill in a controlled environment, such as an assembly plant or a laboratory. In an ill-understood and uncontrolled setting such as outer space (for which the Telerobot was designed), the decision appears well founded. Skeptical architectures [Zerroug and Abramson, 1990] suggest a framework for the cost-effective use of sensory equipment in a (partially) controlled environment. Their coordination scheme appears to hold substantially less promise for applications that involve exploration or navigation.

Equations 11, 12, and 13 relate the probability of a spontaneous occurrence to (i) the costs of sensing, planning, and executing (S, P, E), (ii) the parameter of the PSS (B), and (iii) the severity of difficulties engendered by nature (δ, ϵ). In so doing, they address two key points. First, the intricate interrelationship among the equipment, the search space, and nature-related parameters stresses the importance of careful modeling to the success of any integrated system. Second, the analysis included many simplifying assumptions. Most of them, however, did nothing more than ease the mathematics. The introduction of an heuristic search into the planner, for example, would change very little; it would simply replace all terms that were exponential in B with a different (smaller) function, $f(B)$. This paper's specific results were, of course, highly dependent on its assumptions. Although different architectures, environments, and sets of assumptions would undoubtedly have yielded different formulas, their general qualitative implications should remain. Proponents of other approaches to dynamic planning, then, should use this paper's analysis as a guide to analyses of their own systems and the environments that they encounter.

References

[Abramson, 1990a] B. Abramson. Expected-Outcome: A General Model of Static Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):182–193, 1990.

[Abramson, 1990b] B. Abramson. The Statistics of State-Spaces. *The Annals of Mathematics and Artificial Intelligence*, 2:5–20, 1990.

[Cheeseman, 1988] P. Cheeseman. An Inquiry into Computer Understanding. *Computational Intelligence*, 4(1):58–66, 129–142, 1988.

[Doyle *et al.*, 1986] R. J. Doyle, D. J. Atkinson, and R. S. Doshi. Generating Perception Requests and Expectations to Verify the Execution of Plans. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 81–88, 1986.

[Fikes and Nilsson, 1971] R.E. Fikes and N.J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving . *Artificial Intelligence*, 2:189–208, 1971.

[Georgeff, 1987] M. P. Georgeff. Actions, Processes, and Causality. In *Proceedings of the 1986 Workshop on Reasoning about Actions and Plans*, pages 99–122, 1987.

[Hansson and Mayer, 1990] O. Hansson and A. Mayer. Probabilistic Heuristic Estimates. *The Annals of Mathematics and Artificial Intelligence*, 2:209–220, 1990.

[Horvitz *et al.*, 1988] E.J. Horvitz, J.S. Breese, and M. Henrion. Decision Theory in Expert Systems and Artificial Intelligence. *International Journal of Approximate Reasoning*, 2:247–302, 1988.

[Maes and Brooks, 1990] P. Maes and R.A. Brooks. Learning to Coordinate Behaviors. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 796–802, July/August 1990.

[Ng and Abramson, 1990] K.-C. Ng and B. Abramson. Uncertainty Management in Expert Systems. *IEEE Expert*, 5(2):29–48, 1990.

[Russell and Wefald, 1989] S. Russell and E. Wefald. An Optimal Game-Tree Search Using Rational Meta-Reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1989.

[Wellman, 1990] M.P. Wellman. The STRIPS Assumption for Planning Under Uncertainty. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 198–203, July/August 1990.

[Zerroug and Abramson, 1990] M. Zerroug and B. Abramson. Towards an Architecture for Dynamic Planning. Technical Report 90-05/CS, University Of Southern California, 1990.