# A Survey of the Eighth National Conference on Artificial Intelligence: Pulling Together or Pulling Apart?

Paul R. Cohen

CMPSCI Technical Report 91-68

Experimental Knowledge Systems Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

## Abstract

A survey of 150 papers from the Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90) shows that AI research follows two methodologies, each incomplete with respect to the goals of designing and analyzing AI systems but with complementary strengths. I propose a mixed methodology and illustrate it with examples from the Proceedings.

## 1. Introduction

As fields mature they produce subfields—AI has one or two dozen, depending on how you count. Subfields are differentiated by subject and methodology, by what they study and how they study it. Subfields in AI study intelligent functions such as learning, planning, understanding language, perception; and underpinnings of these functions such as commonsense knowledge and reasoning. We could debate whether it makes sense to study intelligence piecewise—where you solve vision and I solve planning, and someday we get together to build autonomous mobile robots—but this is not our main concern here. If AI researchers are not pulling together, if the field is pulling apart, it is less because we study different subjects than because we have different methods. To support this claim, we present the results of a survey of 150 papers from the *Proceedings of the Eighth National Conference on Artificial Intelligence* (AAAI-90) [0]. We offer evidence for four hypotheses: first, AI research is dominated by two methodologies; second, with respect to the goal of developing science and technology to support the design and analysis of AI systems, neither methodology is sufficient alone; third, the bulk of AI research consequently suffers from familiar methodological problems, such as lack of evaluation, lack of hypotheses and predictions, irrelevant models, and weak analytical tools; and fourth, there exists a methodology that merges the current "big two"and eliminates the conditions that give rise to methodological problems. Our survey provides direct statistical support for the first claim; the other claims are supported by statistical evidence and excerpts from the papers in AAAI-90.

Our presentation has three parts: a summary of the survey and general results, a discussion of our hypotheses, and a pair of appendices that contain details of the survey and statistical analyses. Section 2 briefly describes the 16 substantive questions we asked about each paper. The criteria for answering them are discussed in detail, and illustrated with excerpts from AAAI-90, in Appendix 1. Section 3 paints a picture of AAAI-90 with broad, descriptive statistics. Section 4 restates our hypotheses, and Sections 4.1 – 4.3 discuss the evidence for them (Appendix 2 documents the statistical analyses). Arguments against the methodology we propose in Section 4.3 are considered—but not conceded—in Section 5.

We acknowledge that methodological papers are unpalatable for a variety of reasons. But they also indicate that the field is approaching maturity (e.g., [10] Ch. 7), and so should be welcomed for this reason if not for the problems they raise. In fact, this is an extremely positive paper because, unless we have misread the field very badly, it should be easy to remove the structural, endogenous causes of our methodological problems. Then we have to worry only about conservatism and other sociological impediments, which can be addressed in curricula and editorial policy.

## 2. The Survey

The survey covered 150 of 160 papers from AAAI-90. All the papers were read by one individual, who omitted the ten papers he did not understand or that did not fit easily into Table 1. Each paper is characterized by the 19 fields in Table 1. We will describe these fields only briefly here, in order to get quickly to the survey results (the reader should consult Appendix 1 for detailed descriptions of the fields). Two kinds of data were collected from each paper: the purpose of the research and how the paper convinces the reader that its purpose has been achieved. Fields 3 – 8 of Table 1 represent purposes; specifically, to define models (field 3) prove theorems about the models (field 4) present algorithms (field 5) analyze algorithms (field 6) present systems and/or architectures (field 7) and analyze them (field 8). These purposes are not mutually exclusive; for example, many papers that presented models also proved theorems about the models.

Models are formal characterizations of behaviors (e.g., two papers in AAAI-90 present models of cooperative problem solving) or task environments (e.g., several papers focus on recursive problem-space structures). Some papers extended models to incorporate new behaviors (e.g., extending ordinary constraint-satisfaction problem solving to include dynamic constraints on variables). Some papers generalized models and others differentiated them, demonstrating on the one hand that two or more models have a common core, and, on the other, that a model fails to distinguish behaviors or task environments. Some papers provided formal semantics for models that had previously included vague terms (e.g., probabilistic semantics for costs). More than half the papers in AAAI-90 presented algorithms (field 5) and many also analyzed the algorithms (field 6). Complexity analyses dominated. Surprisingly, only 45 papers presented systems (field 7) and even fewer analyzed systems (field 8). The distinctions between models, algorithms and systems are somewhat subjective and are illustrated in Appendix 1.

Fields 9 – 18 in Table 1 represent methodological tactics for convincing the reader that the purpose of a paper has been achieved. The most common tactic was to present a single example (field 9); but many papers reported studies involving multiple trials, designed to assess performance (field 12), or assess the coverage of techniques on different problems (field 13) or compare performance (field 14). Three fields in Table 1 describe examples and tasks (fields 9, 10,11). *Natural* examples and tasks are those humans encounter, such as natural language understanding, cross-country navigation, and expert tasks; *synthetic* examples and tasks share many characteristics with natural tasks but are contrived (e.g., simulations of robots in dynamic environments); *abstract* examples and tasks are designed to illustrate a single research issue in the simplest possible framework (e.g., N queens, the Yale Shooting problem, Sussman's anomaly, etc.). Some papers described techniques *embedded* in a larger environment (e.g., temporal projection embedded in a planning system).

| | | | | |
|---|---|---|---|---|
| 1. Paper ID number | | | | |
| 2. Paper classification | | | | |
| 3. Define, extend, generalize, differentiate, semantics for models | 72 | | | |
| 4. Theorems and proofs re: model | 49 | | | |
| 5. Present algorithm(s) | 84 | | | |
| 6. Analyze algorithm(s) | 61 | complexity 27 | formal 19 | informal 15 |
| 7. Present system | 45 | | | |
| 8. Analyze aspect(s) of system | 21 | complexity 5 | formal 3 | informal 13 |
| 9. Example type | 133 | natural 39 | synthetic 24 | abstract 70 |
| 10. Task type | 63 | natural 32 | synthetic 9 | abstract 22 |
| 11. Task environment | 63 | embedded 28 | not embed'd 35 | |
| 12. Assess Performance | 38 | | | |
| 13. Assess Coverage | 4 | | | |
| 14. Comparison | 24 | | | |
| 15. Predictions, hypotheses | 25 | | | |
| 16. Probe results | 18 | | | |
| 17. Present unexpected results | 8 | | | |
| 18. Present negative results | 4 | | | |
| 19. Comments | | | | |

**Table 1.** Our classification scheme for AAAI-90 papers. The number of papers in each classification is shown in the columns. For example, of 61 papers that analyzed algorithms, 27 offered complexity analyses, 19 presented other formal analyses, and 15 gave informal analyses. Where possible answers are not listed, the answers are "yes" and "no," and the number of "yes" answers is reported. For example, 18 of the 150 papers probed results. There are no mutually exclusive subsets of fields (although the answers to the question in each field are mutually exclusive) so each paper can contribute to the total for every field.

Relatively few papers presented hypotheses or predictions (field 15). The criteria for what counts as hypotheses and predictions are discussed in Appendix 1, but because the absence of hypotheses in AAAI-90 is central to the rest of this paper, we must note here that worst-case complexity results—which were very common in AAAI-90 papers—did not count as hypotheses or predictions. They *are* predictions of a sort, but predictions of performance in the most extreme circumstances; and they tell us nothing about how common the worst-case circumstances are apt to be, nor how techniques will behave in average cases. Not only were average-case hypotheses and predictions rare, but so too were follow-up experiments to probe previous results (field 16), and reports of negative and unexpected results (fields 17,18). Because hypothesis testing, followup studies and replications with extensions are common and compelling methodological tactics throughout the sciences, their absence from AAAI-90 is troubling.

The survey involved subjective judgments, but no reliability studies have been performed. This caveat and related concerns are discussed further in Appendix 2. To compensate for the lack of reliability, the criteria for classifying the papers are discussed in detail and illustrated with excerpts from the papers themselves in Appendix 1. Excerpts from AAAI-90 are referenced by the following convention: each is identified by a single number which is either the page number in the *Proceedings* on which the excerpt is to be found, or is the page number of the first page of the paper. A few excerpts are unattributed.

## 3. General Results.

Of the 150 papers surveyed, most included one or more examples (field 9) but fewer than half described a task and trials of a system beyond a single example (field 10), and only 45 papers demonstrated performance in some manner (fields 12,13,14). 104 papers offered some kind of analysis (see below). 24 papers probed or otherwise examined results (fields 16,17,18), and 25 papers presented hypotheses or predictions (field 15).

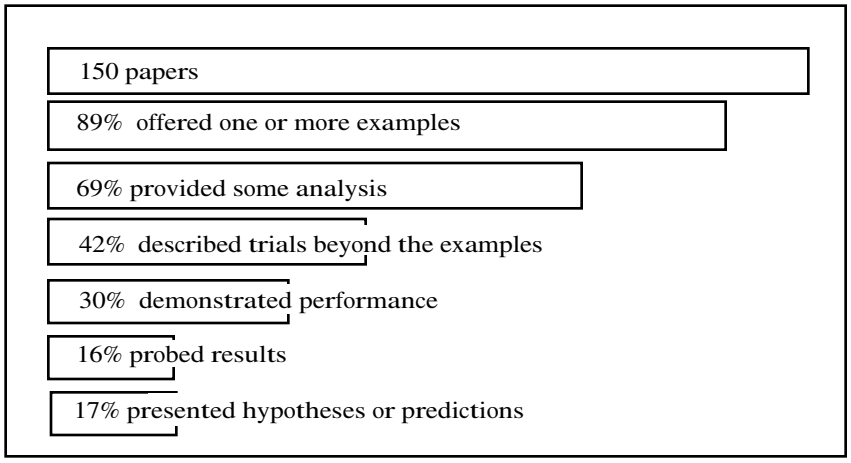



**Figure 1.** Summary of results from the survey of papers in AAAI-90.

These results are summarized in Figure 1. The general picture is that AAAI-90 published preliminary, often unevaluated work. Although one would expect to see hypotheses and predictions even in preliminary research, these were notably absent from AAAI-90.

## 4. Four Hypotheses

*AI is two schools of thought swimming upstream.* — C. R. Beal.

Our survey provides support for four hypotheses about the current state of AI research. First, most AI research is conducted with two methodologies which have in the past been associated with "neat" and "scruffy" styles of AI research. Second, with respect to the goals of providing science and technology to support the design and analysis of AI systems, neither methodology is sufficient alone. Third, common methodological problems arise because AI's methodologies are insufficient to its goals. Fourth, by combining aspects of the two methodologies, we get another less prone to these methodological problems. The following sections discuss the evidence for these hypotheses. Section 4.1 relies heavily on statistical evidence to support the two-methodology hypothesis, whereas Sections 4.2 and 4.3, which discuss the other hypotheses, rely on excerpts from papers in AAAI-90. The third hypothesis, which claims a causal relationship, is supported only indirectly by showing that methodological problems are present when the two methodologies are practiced individually (see Sec. 4.2) and absent when aspects of the two methodologies are combined (see Sec. 4.3). It's an important hypothesis because it claims that many or all of AI's methodological problems have a common root, and, so, it suggests these problems can be corrected en masse. The fourth hypothesis is supported by describing and demonstrating aspects of the combined methodology in some papers in AAAI-90.

### 4.1 Hypothesis 1. Two Methodologies.

To support our first hypothesis—that AI is dominated by two methodologies—we classify the papers in AAAI-90 by some of the fields in Table 1, and show that the classification produces two clusters of papers with few papers in common. Then we demonstrate that the papers in these clusters represent different methodologies, called *model-centered* and *system-centered*, respectively.

We used fields 3 – 8 of Table 1 to classify the papers into three sets and their intersections, shown in Figure 2a. The first set, called MODELS, includes those papers that had "yes" in field 3 or 4, that is, papers that dealt with models. 25 papers dealt with models alone, 43 dealt with models and algorithms, 1 dealt with models and systems, and 4 dealt with all three topics. The second set, called ALGS, includes all papers that presented algorithms (field 5) or some kind of analysis of the algorithms (field 6). The third set, called SYSTEMS, contains papers that presented systems or analyses of systems (fields 7 and 8, respectively). One paper belonged to none of these classes, so the total number of papers for all

further discussions is 149, not 150; this causes some totals in the subsequent analyses to be one less than indicated by Table 1.

The overlap between MODELS and ALGS is considerable, whereas few papers belonged to these classes *and* belonged to SYSTEMS. As shown in Figure 2b, we denote as *model-centered* the papers in MODELS, ALGS and MODELS ∩ ALGS (104 papers in all). We refer to papers from SYSTEMS as *system-centered* (37 papers in all). Eight *hybrid* papers reside in the intersection of the other two classes.

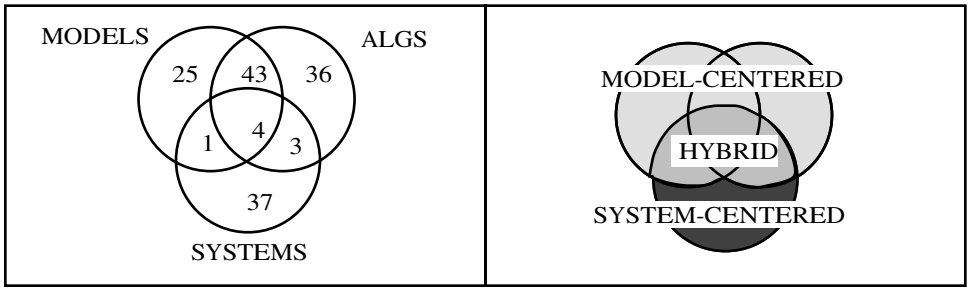



**Figure 2a,b.** Papers in AAAI-90 classified by fields 3 – 8.

Model-centered papers represent one methodology and system-centered papers, another. To show that these methodologies are both real and significantly different, we adopt the following strategy: Starting with the classification of papers in Figure 2b, we test whether the classifications are correlated with methodological choices represented by fields 9 – 18 of Table 1. For example, if most system-centered papers present natural examples and most model-centered papers present abstract examples (field 9), then, because this distribution of task types is unlikely to have occured by chance, the classification of a paper as system-centered or model-centered implies a methodological choice, namely, the choice of an example. Simple statistical tests, described in Appendix 2, tell us whether the methodological choices in fields 9 – 18 are independent of the classifications in Figure 2b. In general, they are not: system-centered papers represent very different methodological tactics than model-centered papers.

The following items describe how system-centered and model-centered papers differ methodologically. Details of the analyses are described in Appendix 2.

1. Model-centered papers present different kinds of examples than system-centered and hybrid papers. In particular, 76% of model-centered papers gave abstract examples or no examples at all, whereas 84% of system-centered and hybrid papers dealt with natural or synthetic examples. This result is highly significant ($\chi^2(6) = 55.5$, $p < .0001$).

2. The classes MODELS, ALGS and MODELS ∩ ALGS (Fig. 2) could not be differentiated by the kinds of examples they contained. 84% of the papers in

MODELS, 81% of the papers in MODELS ∩ ALGS, and 64% of the papers in ALGS, gave abstract examples or no examples at all. Because papers in MODELS, ALGS and MODELS ∩ ALGS presented the same kinds of examples with roughly the same relative frequencies, we are justified in combining the papers in these sets into the single class of model-centered papers. The papers in MODELS are "pre-implementation" and tend to be definitional, whereas those in ALGS and MODELS ∩ ALGS typically describe implemented algorithms. These differences, however, are statistically independent of the kinds of examples that motivate the papers ($\chi^2(6) = 7.26$, $p > .29$).

3. Recall that some papers described *tasks*, that is, multiple trials beyond a single illustrative example. As with examples, tasks are classified as natural, synthetic, abstract, and none (field 10); and as with examples, we find differences between model-centered and system-centered papers in the kinds of tasks they address: 85% of model-centered papers describe abstract tasks or no tasks at all, whereas 58% of system-centered and hybrid papers describe natural or synthetic tasks. This result is highly significant ($\chi^2(6) = 55.4$, $p < .0001$). None of the 25 papers in MODELS addressed a task, which is not surprising if we view them as "pre-implementation" papers; but we were very surprised to find that 41% of system-centered papers described no task, that is, nothing more than a single illustrative example. Still, a greater proportion of system-centered papers (59%) than model-centered papers (33%) described tasks ($\chi^2(2) = 11.9$, $p < .005$).

4. Of the papers that did report multiple trials on tasks, 86% of the system-centered and hybrid papers described embedded task environments, whereas 88% of model-centered papers described non-embedded task environments. Again, this result is highly significant ($\chi^2(2) = 33.9$, $p < .0001$) but hardly surprising: By definition, the techniques discussed in system-centered papers are embedded in a system (otherwise the paper wouldn't have been classified as system-centered in the first place). The surprise is that model-centered papers (mostly from ALGS) were tested so rarely in embedded task environments—in systems or in real physical environments.

5. Model-centered and system-centered papers differ in their orientation toward assessing performance, assessing coverage, and comparing performance (fields 12, 13, 14, respectively). We say that a paper presents a *demonstration* if it reports at least one one of these three activities. Remarkably, a higher proportion of model-centered papers (30%) than system-centered papers (22%) presented demonstrations, even though 25 model-centered papers (from MODELS) were "pre-implementation" papers with nothing to demonstrate. Statistically, this result is suggestive but not significant ($\chi^2(2) = 5.29$, $p < .07$). However, if we look at the papers that described a task (field 10), thereby "declaring their intention" to demonstrate their techniques on multiple trials, and ask the question again, we get a highly significant result: 36% of the system-centered papers that described a task also presented demonstrations, compared with 91% of the model-centered papers and five of the six hybrid papers ($\chi^2(2) = 19.97$, $p < .001$). Even though more

system-centered papers described multiple trials on tasks, relative to model-centered papers, fewer presented successful demonstrations. It seems easier to demonstrate the performance of an algorithm on an abstract problem than an entire system on a natural or synthetic problem. (See Appendix 1 for a list of abstract problems.)

6. Recognizing that demonstrations are only one way to evaluate a technique (and not a particularly informative one, at that) we looked at whether system-centered and model-centered papers had different propensities to *analyze* their contributions. We found that 79% of model-centered papers, six of eight hybrid papers, and just 43% of system-centered papers reported any kind of analysis. This is highly significant ($\chi^2(2) = 16.5$, $p < .0005$); however these results are not strictly comparable with the previous ones because they depend on a slight redefinition of system-centered, model-centered, and hybrid; see Appendix 2.

7. Finally, we looked at the relative frequencies of hypotheses, predictions, probes, unexpected results, and negative results (fields 15 – 18, respectively). We had hoped to analyze these fields separately but only field 15 (hypotheses, predictions) contained enough data to support statistical tests. By combining the fields we were asking whether the researcher had any *expectations* beyond the common assertion that a technique will work (see Appendix 1 for descriptions of fields 15 – 18). Once again, we found a significant effect of methodology: 22% of model-centered, 11% of system-centered, and five of eight hybrid papers had expectations ($\chi^2(2) = 10.5$, $p < .01$). While few papers overall gave evidence of expectations, model-centered and hybrid papers did so more often than system-centered papers, suggesting that the models in model-centered papers may have offered a small advantage in generating hypotheses and predictions.

The paucity of expectations in fields 15 – 18 is disturbing, so we asked whether evidence of expectations could be found in other fields in Table 1. One possibility is field 14, which we used to register papers that compared performance among techniques. We reasoned that the techniques had been selected not arbitrarily but to probe or explore expectations about their relative strengths and weaknesses. Remarkably, model-centered papers numbered 20 of the 24 that compared performance, lending further support to the idea that the models in model-centered papers are used to generate expectations; conversely, lacking models, system-centered papers are generally devoid of expectations.

In summary, we have presented evidence that AI is dominated by two methodologies. Model-centered research involves defining, extending, differentiating and generalizing models, analyzing and proving theorems about these models, designing and analyzing algorithms, and testing algorithms on abstract problems such as N Queens and Blocks World. System-centered research involves designing systems to perform tasks that are too large and multifaceted to be accomplished by a single algorithm. System-centered papers represent different methodological tactics than model-centered papers; they are concerned

with different kinds of examples, tasks, and task environments than model-centered papers. System-centered papers are more apt to describe multiple trials on a task but they are less likely to demonstrate performance than model-centered papers. Systems are less likely to be analyzed than the algorithms in model-centered papers; and system-centered papers present fewer hypotheses, predictions, and other evidence of expectations than model-centered papers. In the crudest terms, system-centered researchers build large systems to solve realistic problems, but without explicit expectations, analyses or even demonstrations of the systems' performance; whereas model-centered researchers typically develop algorithms for simple, abstract problems, but with deeper analysis and expectations, and more demonstrations of success.

## 4.2. Hypotheses 2 and 3: Insufficient Methodologies Cause Methodological Problems.

We are developing a case that comprises four claims: there are two AI methodologies; neither is sufficient, alone; almost nobody is using both methodologies together; however, in combination the methodologies are sufficient. Our results are pretty unequivocal about the first and third claims: the two methodologies are real enough, involving different methodological choices, and only eight of 150 papers bridged the methodologies. This section presents evidence that the methodologies are not sufficient, and the next section argues that a composite methodology *is* sufficient. Along the way we show that common methodological problems—from poor evaluation to absurd assumptions—arise *because* AI's methodologies are not sufficient.

If the goal of AI research is to develop science and technology to support the design and analysis of intelligent computer systems, then neither the model-centered nor the system-centered methodology is sufficient alone. Substitute for "intelligent computer systems" the name of other designed artifacts—airplanes, chemical processes, trading systems, etc.—and one sees immediately that central to our goal is the ability to predict and analyze the behavior of our systems. But our survey shows virtually no interaction between researchers who develop models that are in principle predictive and analytic, and those who build systems. AI has developed a remarkable collection of models; the trouble seems to be that some models are inadequate for predicting and analyzing the behavior of AI systems, while others are not being used in this way.

We can illustrate these points with examples of research that *does* effectively merge model building and system building, research that relies on models to predict the behavior of systems under analysis and systems under design. Tambe and Rosenbloom's contribution to AAAI-90 relies on two kinds of models to discuss issues in the design of production match algorithms. One, the *k-search model* describes the complexity of these algorithms. Tambe and Rosenbloom use this model to show that if productions have a structure called the *unique-attribute* formulation, then a match algorithm will require time linear in the number of

conditions. Thus, they justify the unique-attribute formulation for real-time applications. They report, as several others do in AAAI-90 and elsewhere, that this reduction in complexity is bought at the cost of expressiveness [e.g., 633, 640], the so-called expressiveness/tractability tradeoff [16].

Tradeoffs are essential to designers because they can be used to predict—if only comparatively and qualitatively—the behavior of alternative designs. The most useful tradeoffs are *operational* in the sense of telling the designer what to change—which "knobs to tweak"—to change behavior. The expressiveness/ tractability tradeoff is not operational: it is too general, and researchers have to figure out for themselves how to find a compromise design. Because the model-centered papers in AAAI-90 are not concerned with systems (i.e., they lack architectural knobs to tweak) they do not operationalize the expressiveness/tractability tradeoff (or other tradeoffs); and because these papers do not consider applications, they have no basis for preferring the behavior of one design over another. They say, yes, there is a tradeoff, but until we build a system there's no way to know what to do about it. For example:

> In obtaining generality, our inheritance formalism also becomes intractable. We have tried to keep an open mind on whether it is best to secure a polynomial inheritance algorithm at all costs, or to provide expressive adequacy even if this requires intractable algorithms. ... Both sorts of systems need to be tested. [639]

Tambe and Rosenbloom, however, operationalized the expressiveness/tractability tradeoff by exploring it in the context of production system architectures. This gives them architectural knobs to tweak. They introduce their second model, a framework in which to compare particular kinds of restrictions on the expressiveness of productions (e.g., restrictions on the number of values per attribute). They show that within this framework the unique-attributes formulation is optimal: "All other formulations are either combinatoric, so that they violate the absolute requirement of a polynomial match bound; or they are more restrictive than unique-attributes." [696] Later, they extend the model to incorporate other aspects of the structure of productions, in effect expanding the space of designs by increasing the number of knobs that can be tweaked. In this space, unique-attributes is not guaranteed to be better than other possible formulations.

Like Tambe and Rosenbloom, Subramanian and Feldman develop a model to represent a design tradeoff and to show that some designs are inferior to others:

> [We] demonstrate the conditions under which ... to use EBL to learn macro-rules in recursive domain theories. ... We begin with a logical account of the macro-formation process with a view to understanding the following questions: What is the space of possible macro-rules that can be learnt in a recursive domain theory? ... Under what conditions is using the original domain theory with the rules properly ordered, better than forming partial unwindings of a recursive domain theory? ...

> The overall message is that for structural recursive domain theories where we can find if a rule potentially applies by a small amount of computation, forming self-unwindings of recursive rules is wasteful. The best strategy appears to be compressing the base case reasoning and leaving the recursive rules alone. We proved this using a simple cost model and validated this by a series of experiments. We also provided the algorithm R1 for extracting the base case compressions in such a theory. [949]

Such papers are rare in AAAI-90; only eight of 149 papers reside in the intersection of model-centered and system-centered research. Is this bad? We have offered a couple of examples in which the methodologies are profitably merged, now we document the costs of working in one methodology exclusively. This is most convincing when we let researchers within each methodology speak for themselves. We begin with model-centered research.

### 4.2.1. Models Without Systems

One concern is that the analytical tools of model-centered research do not cut finely enough, and so empirical research is necessary. Worst case complexity analysis—the most common kind of analysis in AAAI-90—does not tell us how systems will perform in practice. Model-centered researchers acknowledge that intractable tasks may in fact be possible and approximations or otherwise weakened models may suffice:

> The worst-case complexity of the algorithm is exponential in the size of the formula. However, with an implementation that uses all possible optimizations, it often gives good results. [166]

> This pessimistic [intractability] result must be taken in perspective. Shieber's algorithm works well in practice, and truly extreme derivational ambiguity is required to lead it to exponential performance. [196]

> Of course complete and tractable subsumption algorithms for the whole language and for the standard semantics presented here cannot be expected. In Allen's interval calculus ... determining all the consequences of a set of constraints is NP-hard. ... That does not render these formalisms useless. On the one hand it remains to be seen to what extent normal cases in practical applications can be handled even by complete algorithms. On the other hand, algorithms for computing subsumption in terminological logics that are incomplete with respect to standard semantics are increasingly being characterized as complete with respect to a weakened semantics. [645]

Another concern is that model-centered research is driven by formal issues that would fade away like ghosts at dawn in light of natural problems. One such argument, by Etherington, Kraus and Perlis [600], suggests that apparent paradoxes in nonmonotonic reasoning disappear when we reconsider what nonmonotonic reasoning is *intended to do*:

> We briefly recount four such "paradoxes" of nonmonotonic reasoning. ... The observed problems can be viewed as stemming from a common root—a misapprehension, common to all the approaches, of the principles underlying this kind of reasoning. ... The *directed* nature of reasoning seems to have been ignored. We contend that the intention of default reasoning is generally not to determine the properties of every individual in the domain, but rather those of some particular individual(s) of interest. ... In the case of the lottery paradox, by considering the fate of every ticket, we face the problem that some ticket must win—giving rise to numerous "preferred" models. If we could reason about only the small set of tickets we might consider buying, there would be no problem with assuming that none of them would win. [601-602]

Even if we agree that an abstract problem is representative of a natural one, solving the former may not convince us that we can solve the latter. Brachman raises this concern in his invited lecture:

> The Yale Shooting Problem and other canonical [nonmonotonic reasoning] problems involve a very small number of axioms to describe their entire world. These may not be fair problems because the knowledge involved is so skeletal. It seems unrealistic to expect a reasoner to conclude intuitively plausible answers in the absence of potentially critical information. By and large, [nonmonotonic reasoning] techniques have yet to be tested on significant "real-world"-sized problems. [1086]

Focusing on practical reasoning tasks not only dispels chimeras, but also guides the search for solutions to formal problems. Shastri points out that reasoning may be intractable, *but we do it*, so we had better figure out how:

> A generalized notion of inference is intractable, yet the human ability to perform tasks such as natural language understanding in real time suggests that we are capable of performing a wide range of inferences with extreme efficiency. The success of AI critically depends on resolving [this] paradox. [563]

Indeed, because the space of extensions and refinements to models is enormous, practical problems must be used to constrain research. For example, Hanks contrasts the general, formal problem of temporal projection with a specific practical projection problem:

> Temporal projection has been studied extensively in the literature on planning and acting, but mainly as a formal problem: one starts with a logic that purports to capture notions involving time, action, change and causality, and argues that the inferences the logic licenses are the intuitively correct ones. This paper takes a somewhat different view, arguing that temporal projection is an interesting *practical* problem. We argue that computing the possible outcomes of a plan, even if formally well-understood, is computationally intractable, and thus one must restrict one's attention to the "important" or "significant" outcomes. This is especially true in domains in which the agent lacks perfect knowledge, and in which

> forces not under the agent's control can change the world, in other words, any interesting domain. [158]

Another reason to merge theoretical and empirical work is that formal models often involve simplifying assumptions, so it is important to check the predictions of the models against practical problems.

> To ensure that the approximations made in Section 2 [do] not invalidate our theoretical results, we compared the iterative-broadening approach to conventional depth-first search on randomly generated problems. [219]

> To a first approximation, we expect symptom clustering to achieve exponential time and space savings over candidate generation. ... However, the exact savings are difficult to determine, because some of the candidates are not minimal and because a candidate may satisfy more than one symptom clustering. Nevertheless, experimental results presented later lend support to a near-exponential increase in performance. [360]

Taken together, these excerpts suggest that in the absence of practical tasks, model-centered research is prone to several methodological problems. It is evidently possible to work on formal problems that may not arise in practice, to lose track of the purpose of a kind of reasoning, to not exploit practical constraints when designing solutions to formal problems, and to solve formal problems without checking one's assumptions or simplifications in practical situations. How common are these pathologies? It is difficult to tell because they show up primarily when a researcher attempts to use models in systems, which is extremely rare in AAAI-90. However, we do know that virtually all model-centered papers are prone to these problems. Consider: 76% of model-centered papers gave abstract examples or no examples; only 33% of these papers described tested implementations, and more than half of these were tested on abstract problems; only four model-centered papers described techniques embedded in larger software or hardware environments.

### 4.2.2. Systems Without Models

"Look Ma, no hands" — J. McCarthy.

Model-centered research at least produces models, proofs, theorems, algorithms, and analyses. It is difficult to say what exclusively system-centered research produces. In general, system-centered papers are descriptive rather than analytic; they describe systems that *do* things, such as distributed problem solving, diagnosis, design, and so on. It is either tacitly assumed or vaguely asserted that something is learned or demonstrated by implementing and "testing" the systems described in these papers; for example,

> We have implemented the projector and tested it on fairly complex examples...

> To investigate the performance of this implementation of our protocol...
>
> We have tested our prover on some problems that are available in the theorem-proving literature.

Lacking a clear statement in the system-centered papers of why one should build systems, we turned to Lenat and Feigenbaum's discussion of their *empirical inquiry hypothesis*:

> Compared to Nature we suffer from a poverty of the imagination; it is thus much easier for us to uncover than to invent. Premature mathematization keeps Nature's surprises hidden. ... This attitude leads to our central methodological hypothesis, our paradigm for AI research:
>
> Empirical Inquiry Hypothesis: Intelligence is still so poorly understood that Nature still holds most of the important surprises in store for us. So the most profitable way to investigate AI is to embody our hypotheses in programs, and gather data by running the programs. The surprises usually suggest revisions that start the cycle over again. Progress depends on these experiments being able to falsify our hypotheses; i.e., these programs must be capable of behavior not expected by the experimenter. [15]

Apparently the methodology is not being practiced by system-centered researchers or is not producing the desired results. Our survey tells us that in general neither model-centered nor system-centered researchers "embody hypotheses in programs," or "gather data by running the programs." In fact, only 25 papers presented hypotheses that could surprise the experimenter and only two of these were system-centered (the rest presented the hypotheses that a program works, or works better than another program, or presented no hypothesis at all). And if Nature is so full of surprises, why did only 24 papers report negative results, unexpected results, or probe results?

One is tempted to criticize these papers, as Lenat and Feigenbaum do, as "using the computer either (a) as an engineering workhorse, or (b) as a fancy sort of word processor (to help articulate one's hypothesis), or, at worst, (c) as a (self-) deceptive device masquerading as an experiment." [15, p.1177] In other words, the empirical inquiry hypothesis is ok but AI researchers are not. But we believe there is something inherently wrong with the empirical inquiry hypothesis and with system-centered research in general: How can a system exhibit "behavior not expected by the experimenter" if there are no expectations, and how can there be expectations without some kind of predictive model of the system? One needn't subscribe to formal, mathematical models, but nor can one proceed in hope of being surprised by Nature. The empirical inquiry hypothesis should say, but does not, that hypotheses and expectations are derived from models—formal or informal—of the programs we design and build.

We will argue later that the lack of models in system-centered research is the distal cause of a host of methodological problems. The proximal cause is the

reliance on demonstrations of performance. Many researchers apparently believe that implementing systems is both necessary and sufficient to claims of progress in AI. Whereas necessary is debatable, sufficient is dead wrong. First, although statements of the form "my system produces such-and-such behavior" (abbreviated S → B) are sometimes called "existence proofs," nobody ever claimed that these programs could *not* exist—no hypothesis or conjecture is being tested by implementing them. S → B is not itself a hypothesis. Neither S → B nor its negation are practically refutable: tell any hacker that a system cannot be made to do something and it's as good as done. In fact, the *only* empirical claim made of these systems is that they exist; all other claims are vague and promissory. For example, "We presented a sketch of an architecture ... that we believe will be of use in exploring various issues of opportunism and flexible plan use." Very few systems merit attention on the basis of their existence, alone.

Second, desired behaviors are specified very loosely (e.g., real-time problem solving, graceful degradation, situated action) and so S → B is less a hypothesis than a definition: B is the behavior produced by S. The "wishful mnemonic" approach to system design and software engineering, excoriated by McDermott in 1976 [18], continues unabated today. Behaviors are what are produced by the components of systems that carry the names of behaviors (e.g., "scheduling" is what the "scheduler" does). This transference is exhiliarating—we can build anything we can imagine and call it anything we like. The downside is that what we *can* build crowds out what we *need to* build to produce particular behavior in a particular environment.

Third, demonstrating that S → B does not mean that S is a particularly *good* way to produce B. Lacking such an assurance, we can only conclude that S works adequately but its design is unjustified. Occasionally, a researcher will demonstrate that one program works better than another, but in system-centered research the result is rarely explained.

Fourth, demonstrations don't tell us *why* a system works, what environmental conditions it is sensitive to, when it is expected to fail, or how it is expected to scale up—in short, demonstrations don't amount to understanding [3, 6, 7, 8, 13]. And, finally, implementing something once doesn't mean we learn enough to repeat the trick. If all AI systems are one-off designs, and the only thing we learn about each is that it works, then the "science of design" of AI systems will be a long time coming.

These methodological problems have a common root: system-centered researchers rarely have models of how their systems are expected to behave. Designing and analyzing *any* complex artifact without models is very difficult; imagine designing bridges without models of stress and deflection; or hulls without models of fluid flow; or drug therapies without models of metabolism and other physiological processes. Yet with few exceptions, described below, system-centered papers in AAAI-90 lacked models. Given this, methodological problems

are unavoidable. Lacking models of how systems are expected to behave, we will see no predictions, no hypotheses, no unexpected results or negative results; only assertions that a system "works." Conversely, models define behaviors, avoiding McDermott's wishful mnemonic problem. Models provide exogenous standards for evaluating performance, bringing objectivity to the claim that a system works. And models can represent causal influences on performance, allowing us to predict performance and test hypotheses about why systems perform well or poorly in particular conditions. Models that serve this purpose—predicting and explaining performance—are *necessary* if a system is to contribute to the science of AI, to be more than, in Lenat and Feigenbaum's words, "an engineering workhorse, a fancy sort of word processor, or a (self-) deceptive device masquerading as an experiment."

### 4.2.3. Models and Systems Together

Given these arguments it should not be surprising that models are common among system-centered papers that *do* test hypotheses or explain behavior. An excellent example is Etzioni's explanation in terms of nonrecursive problem space structure of why PRODIGY/EBL works:

> I formalized the notion of nonrecursive explanations in terms of the problem space graph (PSG)...PRODIGY/EBL's nonrecursive explanations correspond to nonrecursive PSG subgraphs. ... I demonstrated the practical import of this analysis via two experiments. First, I showed that PRODIGY/EBL's performance degrades in the augmented Blocksworld, a problem space robbed of its nonrecursive PSG subgraphs. Second, I showed that a program that extracts nonrecursive explanations directly from the PSG matches PRODIGY/EBL's performance on Minton's problem spaces. Both experiments lend credence to the claim that PRODIGY/EBL's primary source of power is nonrecursive problem space structure. [921]

Minton, Johnston, Philips and Laird [23] ran experiments to explain why a particular neural network performs so well on constraint satisfaction problems, and subsequently incorporated the results of this analysis into a scheduling algorithm for, among other things, space shuttle payload scheduling problems. Based on a probabilistic model, they were able to predict the circumstances under which the algorithm would perform more or less well.

Pollack and Ringuette [183] explored a filtering mechanism that "restricts deliberation ... to options that are compatible with the performance of already intended actions." In one experiment they test the hypothesis that the filtering mechanism improves performance. Unlike most experiments in AAAI-90, Pollack and Ringuette's carefully varied the experimental conditions, and, consequently, revealed a tradeoff between the conditions (in this case, the rate of change in the environment) and performance. This led to several hypotheses, each derived from a causal model relating environmental conditions, architecture structure, and behavior. Note that Pollack and Ringuette's strategy of varying

environmental conditions made sense only because they had a hypothesis about the relationships between the conditions and performance; otherwise they would just have been aimlessly tweaking conditions in the hope that Nature would deliver a surprise.

Clearly, models do not have to be quantitative; in the last example they were qualitative and causal. Moreover, models can be developed as post-hoc explanations in service of future design efforts, as in Etzioni's analysis and Minton et al.'s work; or they can evolve over a series of experiments such as Pollack and Ringuette's. The important thing is that these models support the design and analysis of AI systems; they are crucial to answering the questions asked by every designer: how does it work? when will it work well and poorly? will it work in this environment?

### 4.3. Hypothesis 4. There Exists a Sufficient Methodology.

Here we document the evidence in AAAI-90 of a methodology sufficient to the goals of providing science and technology to support the design and analysis of AI systems. We call the methodology MAD, for Modelling, Analysis and Design. MAD involves seven activities:

- assessing environmental factors that affect behavior;
- modelling the causal relationships between a system's design, its environment, and its behavior;
- designing or redesigning a system (or part of a system);
- predicting how the system will behave;
- running experiments to test the predictions;
- explaining unexpected results and modifying models and system design;
- generalizing the models to classes of systems, environments and behaviors.

None of the papers in AAAI-90 reported all these activities, not even the system-centered papers cited earlier which relied successfully on models. Thus, it is worth discussing the MAD activities in some detail, illustrating each with examples from AAAI-90.

**Environment assessment.** To build a predictive model of how systems will behave in a particular environment, we must decide which aspects of the environment to include in the model and how accurately they must be represented. Interestingly, examples of environment assessment are rare in AAAI-90. They include fairly vague characterizations, such as, "Our system...enables users to learn within the context of their work on real-world problems," [420] as well as fairly precise requirements placed by the environment on the system, such as "when designing our current media coordinator [we] showed that people strongly prefer sentence breaks that are correlated with

picture breaks." [442] Many papers focused on a single aspect of environments. Time [e.g., 132, 158] and the recursive structure of problem spaces [e.g., 916, 336, 942] are examples. Only one paper explicitly intended to study the influences upon design of several, interacting aspects of an environment—to seek "an improved understanding of the relationship between agent design and environmental factors." [183]

Environment assessment produces assumptions about the environment; for example, one might assume that events are generated by a Poisson process, or that actions are instantaneous, or that a sentence contains redundant components. These assumptions say, for the purposes of designing and analyzing systems for *this* environment, it probably won't hurt to simplify the characterization of the environment. Assumptions were plentiful in AAAI-90, especially in the model-centered papers, but they were assumptions about no *particular* environment; and, we sometimes suspected, about no plausible environment. This is where the rift between model-centered and system-centered research begins: the assumptions that underlie models often preclude their application to the design and analysis of systems. One way to close the rift is to ground research in a particular environment—to make environment assessment a regular feature of the research. This needn't preclude generality: we can still build models for the entire class of environments of which *this* one is representative, and we will be spared basing our models on assumptions that cannot hold in any environment. (Another way to close the rift is to test the sensitivity of a system to violations of the assumptions; see "Experiments," below).

**Modelling.** Models support all the MAD activities: design, prediction, experimentation, explanation, and generalization. To support these activities models must answer two questions:

1. If we change the design of a system, how will behavior be affected?
2. If we change environmental conditions, how will behavior be affected?

Models come in many varieties, from simple qualitative relationships to fairly precise functional relationships. Tambe and Rosenbloom, for example, develop a qualitative model to show that the unique-attributes design is the best possible within a particular design space, but is inferior in an extended design space [696]. They are among the few authors who attempt to answer question 1, above. Minton et al. [23] give the following probability that the min-conflicts heuristic will make a mistake assigning a value to a variable:

$$\Pr(\text{mistake}) \leq (k - 1)\, e^{-2(pc - d)^2/c}$$

The important thing about this model is that it relates the probability of a behavior (making mistakes) to measurable characteristics of the problem-solver's search space (the terms on the right of the inequality). Thus, Minton et al. can

predict behavior and, as they do in their paper, explain why the min-conflicts heuristic performs so well. Characterizing the search space was the most common tactic for answering question 2, above; for example, Etzioni [916] and Subramanian and Feldman [942] focused on the recursive structure of problem spaces to predict and explain problem-solving behavior. Unfortunately, many models in AAAI-90 gave only qualitative, worst-case characterizations of search spaces (i.e., intractability results) which could not be used to answer either of the questions, above. We did not classify the kinds of models developed in AAAI-90, but the paucity of hypotheses and predictions among the papers suggests either that the models were for some reason not being used to answer questions 1 and 2, above, or, more likely, were not intended to answer the questions. It seems likely that most of the models described in AAAI-90 cannot support most MAD activities.

**Design and Redesign**. Designs, or rather sketches of designs, abound in AAAI-90, especially in system-centered papers. Most are presented without explanation or justification—here's what we are trying to build, here's how we did it. The MAD methodology aims to justify design decisions with models. In *top-down* design, one first derives models and designs from them; Dechter, for instance, clearly intends her models to be used this way:

> A formal treatment of the expressiveness gained by hidden units ... [is] still not available. ... Our intention is to investigate formally the role of hidden units and devise systematic schemes for designing systems incorporating hidden units. [556]

Alternatively, models are developed at the same time as designs. This is an incremental version of MAD, in which designs or parts of designs are implemented to provide empirical data, which flesh out models, which become the basis for redesign. For example, Pollack and Ringuette [183] expected to find a functional relationship between the cost and benefit of filtering in different environmental conditions, but they did not know its form until they ran an experiment (Exp. 2, p. 188). They discovered that the benefits of filtering did not warrant its costs in any conditions, but the ratio of benefit to cost increased with the rate of change of the environment. They knew that as the run time of tasks increased, so would the benefit of filtering those tasks; and, they assumed, so would the accuracy of the results. On the basis of this qualitative model, they proposed to change their design "to implement more accurate (and costly) deliberation mechanisms in the near future. For these, filtering may be much more valuable." [188] This paper is one of a small handful in AAAI-90 that justify design revisions based on models; another excellent example is de Kleer's revisions to the design of TMSs to exploit locality in the underlying structure of some problems [264].

**Prediction**. Prediction is central in the MAD methodology: you predict how a system will behave during the design process; you test predictions in experiments; you explain the disparities between predictions and reality after the experiments; and when you generalize a predictive model, you attempt to preserve as much

predictive power as possible, even as the range of environmental conditions, design decisions, and behaviors increases. Prediction is our criterion for *understanding* a system: we can claim understanding when we can predict with some degree of success how changes in design or changes in environmental conditions will affect behavior.

Without predictions it is virtually impossible to evaluate a system; all one can do is demonstrate that the system works more or less well. If you want to know why it works, or when it is likely to break, you need a model. For example,

> If repairing a constraint violation requires completely revising the current assignment, then the min-conflicts heuristic will offer little guidance. This intuition is partially captured by the [previous] analysis [see the discussion of Pr(mistake), above]... which shows how the effectiveness of the heuristic is inversely related to the distance to a solution. [23]

The MAD view of prediction is very pragmatic: it rejects the abstract argument that prediction is impossible in principle, taking instead the view that even crude, qualitative, somewhat inaccurate predictions can serve designers in practice, especially when incorporated into an iterative cycle of design, experiments, explanations, and redesign (see Sec. 5).

**Experiments.** Experiments have three main purposes in the MAD methodology: to test predictions, to probe models, and to discover behaviors. The first two are directed, the third is exploratory. In AAAI-90, very few experiments served these purposes; instead they demonstrated performance. While this contributes little to our understanding of our systems, if we are going to keep doing it, we should at least develop meaningful, efficient measures of performance. Not surprisingly, this too can be profitably guided by models. For example, Fayyad and Irani ask,

> Suppose one gives a new algorithm for generating decision trees, how then can one go about establishing that it is indeed an improvement? To date, the answer ... has been: Compare the performance of the new algorithm with that of the old algorithm by running both on many data sets. This is a slow process that does not necessarily produce conclusive results. On the other hand, suppose one were able to prove that given a data set, Algorithm A will always (or 'most of the time') generate a tree that has fewer leaves than the tree generated by Algorithm B. Then the results of this paper can be used to claim that Algorithm A is 'better' than Algorithm B. [754]

In short, they derive from a model the result that the number of leaves in a tree is a proxy for many other performance measures, so instead of comparing performance directly, we can compare leafiness. Most performance measures in AAAI-90 are not so carefully justified. Eskey and Zweben point out that a common performance measure—run-time speed up—is *not* a proxy for the measure that represents their goals as designers, and so should not be adopted without careful consideration [908]. The correlation between run-time speed up and the measure

they prefer (see their Tables 2 and 3) is only .26. Researchers who select run-time as an "obvious" performance measure should not expect it to correlate with anything they care about.

Experiment designs are informed by models. Models describe how behaviors are affected by factors in the environment and system design parameters; and experiments test these causal hypotheses. Models tell us where to look for results. For example, although the following excerpt did not herald an experiment, it does suggest where to look for an effect —in "borderline situations"—if an experiment was run:

> Surprisingly, ... there might exist a semi-cooperative deal that dominates all cooperative deals and does not achieve both agents' goals. It turns out this is a borderline situation [104]

This much is recognizable as the conventional "hypothesis testing" view of experiments: a model makes predictions about how changes in the environment or changes in design will affect behavior, and an experiment tests the predictions. But pick up a typical text on experiment design and analysis, and you are unlikely to find any discussion of a subtler, more important relationship between models and experiments: Just as experiment designs are informed by models, so too are models informed by experiment results. Sometimes, results will contradict predictions, but often they will flesh them out, providing data to replace rough, qualitative models with functional, quantitative ones. This iterative, exploratory development of models is described in a recent paper by Langley and Drummond, who see it as the future not only of individual research projects but of the entire field of experimental AI:

> In the initial stages, researchers should be satisfied with qualitative regularities that show one method as better than another in certain conditions, or that show one environmental factor as more devastating ... than another. ... Later stages ... should move beyond qualitative conclusions, using experimental studies to direct the seach for quantitative laws that can actually predict performance in unobserved situations. In the longer term, results of this sort should lead to theoretical analyses that explain results at a deeper level, using average-case methods rather than worst-case assumptions. [13, p. 113]

Langley and Drummond's paper raises many issues in experiment design, including the use of benchmarks. Lately, the calls for benchmarks and common experimental environments have increased in frequency and intensity; for example, DARPA recently sponsored a workshop on benchmarks and metrics and is instituting benchmarks in some of their research programs. We believe that benchmarks and common environments address a symptom—the lack of evaluation of systems—not its cause, and, worse, divert attention from the cause. The principal reason that we don't run experiments in AI is that we don't have hypotheses to test. Instituting benchmarks won't increase the number of hypotheses, only the number of demonstrations of performance [6]. This states the

case too strongly—benchmarks can certainly provide common evaluation criteria and *may* provide the impetus for researchers to understand why their systems perform poorly[1]—but we shouldn't think that instituting benchmarks will fix AI's methodological problems, particularly the lack of predictions and hypotheses.

Nor should we think that common experimental environments will provide us that most elusive of scientific criteria, replicability. It is claimed that if we all perform our experiments in the same "laboratory," (i.e., the same software testbed) then the results will be comparable, replicable, and cumulative.[2] Like the call for benchmarks, this isn't a bad idea but it diverts attention from a real methodological problem. Replicability in other fields is not the replicability of laboratories but the replicability of *results across* laboratories. The strongest results are the ones that hold up in many different environments. If we say that AI systems are so complex that we cannot hope to replicate results across systems, and so for the sake of comparability and cumulativity we should work in a single, common system, then we are by this device diverting attention from a fundamental problem: we understand our techniques so poorly that we cannot say which aspects of their behavior should be replicable in different systems and environments. The solution is to build models that predict behavior; these predictions should then be replicable in all systems and environments that are described by the models.

In sum, experimental work without models is only half a loaf. We can fiddle with the parameters of our systems to "see what happens"; we can demonstrate performance on benchmarks; we can compare techniques within a common experimental environment. All of these are valuable exploratory techniques. All are preferrable to unsubstantiated claims of success. But none is half as convincing as a test of a prediction derived from a model and replicated across environments.

**Explanation**: By explanation we mean accounting for data; for example, Minton et al. account for the performance of the min-conflicts heuristic with the model described above. On the other hand, we may have to explain why data do not support predictions. For example, Hirschberg discovered problems with her model of which features of speech predict stress (accent) assignment: "Even from such slim data, it appears that the simple mapping between closed-class and deaccentuation employed in most text-to-speech systems must be modified." [955] Explanation of incorrect predictions leads to revisions. In Hirschberg's case and the natural sciences in general, incorrect predictions lead to revisions of models. However, the behaviors of AI systems are artificial phenomena, so if models make incorrect predictions about behaviors, should we revise the models or the systems?

---

[1]Mark Drummond pointed this out.

[2]Raj Reddy and other panelists at the recent DARPA Workshop on Planning, Scheduling, and Control made this claim.

This question recently arose in our Phoenix system [4,5,9]. On the basis of a model we predicted that problems would be solved most efficiently in a particular order; however, the prediction failed—performance was very inefficient in one of four experimental conditions. Our model included terms that represented the problem-solving environment and it also made some assumptions about the problem-solving architecture. To explain our results we first showed that the model correctly characterized the environment, and then we attributed the failed prediction to one of these assumptions. This raised an interesting question: If a model predicts that given an assumption about the design of a system, performance should be better than it actually is in experiments, then should we modify the model or redesign the system to conform to the assumption? Modifying the model serves no purpose besides formalizing a bad design; the right answer is to modify the design to bring it in line with the model.

**Generalization**: Whenever we predict the behavior of one design in one environment, we should ideally be predicting similar behaviors for similar designs in related environments. In other words, models should generalize over designs, environmental conditions and behaviors. Model-centered and system-centered researchers have different views of generality: the former has a general model, the latter has a specific system, and neither moves an inch toward the other. The laurels would seem to go to the model-centered researcher, except that the innovations of the system-centered researcher may generate dozens or hundreds of imitations, re-implementations and improvements. Eventually, someone writes a paper that states generally and more or less formally what all these systems do; for example, Clancey's heuristic classification paper [1], Mitchell's characterization of generalization as search [19], and Korf's paper on planning as search [12]. The trouble is that such papers are *rare*.

The activities just discussed can be combined to yield several styles of AI research. We mentioned hypothesis testing, where predictions are generated from models and tested empirically in systems. We also mentioned exploratory model development, where empirical work is intended to first suggest and then refine models [13]. Sometimes, explanation of behavior in terms of models is the principal goal. Sometimes the goal is to design a system or a component of a system, given models of how the artifact will behave in a particular environment. Long-term, large-scale projects will emphasize different MAD activities at different times. For example, in our Phoenix project it was clearly impossible to design in a top-down fashion—from nonexistent models—the architecture of Phoenix agents. (These agents are embedded in simulated bulldozers and firebosses and plan how to fight simulated forest fires [5].) Instead, we differentiated *fixed* design decisions, which will not be reviewed anytime soon; *reviewable* decisions which are reviewed after they are implemented and after models are developed to support the analysis; and *justifiable* decisions, which are based in models before being implemented. This division enabled us to get Phoenix up and running, providing us with an

empirical environment in which to develop models iteratively, make predictions, review design decisions in light of new models, propose new design decisions, and explain performance. To date, most of our modelling effort has been aimed at analyzing reviewable design decisions; for example, although Phoenix agents currently work on multiple fires simultaneously, we have recently developed a model that suggest this is not the best use of resources. If the model holds up empirically, then we will revise the design decision. In sum, although the MAD activities get "mixed and matched" at different stages of a research project, the constant theme is a commitment to develop or adapt models to support the analysis and design of systems.

## 5. Anticipating Arguments Against MAD

Here we consider five arguments against the MAD methodology, and, more generally against any attempt to base the design and analysis of AI systems in models. We do not believe these arguments, we present them to refute them.

### "Predictive Models of AI Systems are Unachievable"

As we work with more complex environments, and with architectures that produce complex behaviors from interactions of simpler behaviors, the goal of developing models to predict behavior seems increasingly remote. Some researchers claim that behavior is in principle unpredictable, so the only way to design systems is as Nature does, by mutation and selection (e.g., [14], p. 25). A related argument is that AI systems are too complex to be modelled in their entirety. But, in fact, complex systems *can* be modelled and behavior can be predicted—if not accurately, at least accurately enough to inform design. Particularly useful, as we noted earlier, are models of design tradeoffs. These need not be accurate, they might be only qualitative, but they help designers navigate the space of designs. Moreover, once a prototype design is implemented, even qualitative design tradeoffs can quickly be enhanced by empirical data. Nor is it necessary to model an entire system to predict its performance. By modelling a critical component of a system—a bottleneck, perhaps—one can predict the gross behavior of an entire system. So the question is not whether predicting behavior is possible in principle, nor whether it is possible to model an entire, complex system, but whether predicting the behavior of important components of systems is useful in practice.

### "Predictive Models Lead to Predictable, Boring Systems"

Another kind of argument goes like this: just how intelligent is a predictable AI system? How do we reconcile the desire for predictability with the desire to be surprised by an AI system? These questions raise some fundamental issues about the nature of novel, creative reasoning, questions that we cannot address here for

want of space and expertise[3] . But we will say this: most of what we mean by creativity involves relatively small variations on a theme; new themes are introduced very infrequently. Nothing in MAD precludes designing a system that is predicted to produce novel variations on a theme. No individual variation would be predictable, but nor would the system stray from the theme.

## "Premature Mathematization Keeps Nature's Surprises Hidden"

Another possible argument against MAD is that modelling discourages exploration or, as Lenat and Feigenbaum put it, "Premature mathematization keeps Nature's surprises hidden." We know of no area of inquiry that has been retarded by efforts to build formal models *of Nature*, but obviously our understanding of Nature—expressed formally or informally—is not advanced by mathematization that has only the most tenuous connection to Nature. Some of Brachman's comments can be interpreted as voicing this concern:

> More theorems and proofs than ever have appeared in recent KR [knowledge representation] papers and the body of mathematics in support of KR has grown dramatically. A formal semantics is now an obligatory accompaniment of the description of a novel KR system. The tremendous upsurge in KR theory has seemingly come at the expense of experimentation in the field...But the pendulum may have swung too far, inadvertently causing a rift between the formalists and those concerned with applications, and causing less and less of the KR literature to have any impact on the rest of AI and on practice. [1085]

There should be no possibility in MAD of the mathematical tail wagging the system-designer's dog. The goal of MAD is to design and analyze systems with the help of models, and to develop new models when the available ones are not sufficient for the purposes of design and analysis of systems. Models serve design and analysis. The methodology simply does not endorse modelling for its own sake.

## The Synchronization Problem

Another potential argument against MAD is an apparent *synchronization* problem: system-centered researchers often find that model-centered researchers provide formal accounts of things that they (the system-centered researchers) have assumed all along. Probabilistic accounts of certainty factors came along a decade after MYCIN [11]; semantics for STRIPS operators were developed later yet [17]. The synchronization problem is that system-centered researchers don't get models when they need them—during design and analysis of systems. We believe the

---

[3]Nort Fowler brought these questions to our attention. They are addressed in Margaret Boden's The Creative Mind, forthcoming from Basic Books.

problem is real, but we believe that MAD alleviates it by encouraging the simultaneous development of models and systems.

## "MAD Misinterprets the Purpose of AI"

Finally, MAD might be rejected on the grounds it misinterprets the purpose of AI. Matt Ginsberg recently put it this way: "You think AI has to do with designing and analyzing systems; I think AI is like medieval chemistry: Design anything you like to try to turn lead into gold, but you won't succeed until you invent nuclear physics. AI theorists are trying to invent nuclear physics. Systems are premature." Paring away the challenges to any given aspect of this analogy, one is left with a basic dispute about how to proceed in AI. Model-centered researchers will say that systems are premature lacking formal models of intelligence. System-centered researchers will say models are superfluous because the goals of AI are satisfied if we can build systems that work, and this can be accomplished without models. Unless we are willing to dismiss one group or the other as wrong about the proper goals and methods of AI, we have to believe both. We have to believe that the goals of AI are to build formal models of intelligence *and* to build intelligent systems. The only question is whether these should be the activities of different cadres of researchers, as they are now, or should be merged, somehow. The symbiosis between the activities is obvious: with models we can design and analyze systems, predict their performance, explain deviations from performance, and so on; with systems we can test the assumptions of models, focus on models for tasks that actually exist, revise the models in response to empirical data, and so on. MAD doesn't misinterpret the goals of AI, it provides a necessary framework in which to achieve them simultaneously.

## Acknowledgments

I thank many colleagues for spirited discussions and comments: Carole Beal, Nort Fowler, Pat Langley, Mark Drummond, Matt Ginsberg, and Glenn Shafer (who challenged me to characterize AI's methodologies); and Adele Howe, Cynthia Loiselle, Scott Anderson, Dave Hart, and other members of the EKSL. I am especially grateful to Alan Meyrowitz at the Office of Naval Research, for intellectual and financial support during my sabbatical, which led to my first experiments within the MAD methodology and to this work.

## References

[0] AAAI–90. (1990). *Proceedings of the Eighth National Conference on Artificial Intelligence.* July 29 – August 3, 1990. Boston, Mass. AAAI Press/The MIT Press.

[1] Clancey, W. Heuristic Classification. *Artificial Intelligence*. (27) 289 – 350. 1985.

[2] Cohen, P.R. Designing and Analysing Strategies for Phoenix from Models. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*. K. Sycara (Ed.) San Mateo, CA.: Morgan-Kaufmann, Inc. pp. 9 – 21.

[3] Cohen, P. R.. Evaluation and Case-based Reasoning. *Proceedings of the Second Annual Workshop on Case-based Reasoning*, Pensacola Beach, FL. May 30—June 2, 1989. pp. 168—172.

[4] Cohen, P. R. Methodological Problems, a Model-based Design and Analysis Methodology, and an Example. *Proceedings of the International Symposium on Methodologies for Intelligent Systems*. pp. 33 – 50. Knoxville, Tennessee, Oct. 25-27, 1990.

[5] Cohen, P. R., Greenberg, M. L., Hart, D.M., Howe, A. E. Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments. *AI Magazine*. **10**(3): 32-48, 1989.

[6] Cohen, P. R. and Howe, A.E. Benchmarks are not enough; Evaluation metrics depend on the hypothesis. Presented at the *Workshop on Benchmarks and Metrics*, Moffett Field, CA, June 24, 1990

[7] Cohen, P. R. and Howe, A.E. How evaluation guides AI research. *AI Magazine*. **9**(4): 35 - 43, 1988.

[8] Cohen, P. R. and Howe, A.E. Toward AI research methodology: Three case studies in evaluation. *IEEE Transactions on Systems, Man and Cybernetics.* **19**(3): 634-646, 1988.

[9] Adele E. Howe, David M. Hart, and Paul R. Cohen. Addressing Real-Time Constraints in the Design of Autonomous Agents, *The Journal of Real-Time Systems*, Vol. 1, pp.81-97, 1990.

[10] De Mey, M. (1982) The Cognitive Paradigm. Boston: D. Reidel.

[11] Heckerman, D. (1986) Probabilistic interpretations for MYCIN's certainty factors. In Uncertainty in Artificial Intelligence, L. Kanal and J. Lemmer (Eds.) North-Holland. pp. 167 – 196.

[12] Korf, R. Planning as search: A quantitative approach. *Artificial Intelligence* 33, 65 – 88. 1987.

[13] Langley, P. and Drummond, M. (1990) Toward an experimental science of planning. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control*. K. Sycara (Ed.) San Mateo, CA.: Morgan-Kaufmann, Inc. pp. 109 – 114.

[14] Langton, C. Artificial Life. Santa Fe Institute Studies in the Sciences of Complexity. 1989.

[15] Lenat, D. B. and Feigenbaum, E. A. On the Thresholds of Knowledge. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*. 1173--1182, 1987.

[16] Levesque, H. J. and Brachman, R. J. (1985) A fundamental tradeoff in knowledge representation and reasoning (Revised version). In *Readings in Knowledge Representation*. R.J. Brachman and H.J. Levesque (Eds.) San Mateo, CA.: Morgan-Kaufmann, Inc. pp. 41 – 70.

[17] Lifschitz, V. (1987) On the semantics of STRIPS. In *Reasoning About Actions and Plans*, M. Georgeff and A. Lansky (Eds.) San Mateo, CA.: Morgan-Kaufmann, Inc. pp. 1 – 9.

[18] McDermott, D. Artificial Intelligence meets Natural Stupidity. In Mind Design. J. Haugeland (ed.). pp. 143 – 160. Montgomery, VT.: Bradford. 1981.

[19] Mitchell, T.M. Generalization as search. In *Readings in Artificial Intelligence.* B. L. Webber and N. J. Nilsson (Eds.) Tioga Press, 1981.

## Appendix 1. The Fields in Table 1.

### Fields 3 and 4: Define, extend, generalize, differentiate, semantics for models; and theorems and proofs re: model

Many papers focused on models of reasoning. The word model is used many ways in AI, but we intend it to mean an abstract, typically formal description of behavior and/or environmental factors or design decisions that affect behavior. The purpose of building a model is to analyze its properties, assuming (often implicitly) that they will carry over to systems that implement the models. For example,

> An important area of research is to devise models of introspective reasoning that take into account resource limitations. Under the view that a KB is completely characterized by the set of beliefs it represents ... it seems natural to model KBs in terms of *belief*. ... The best understood models of belief are based on possible-world semantics. ... Unfortunately, [these models assume] a property often referred to as *logical omniscience*, which renders reasoning undecidable in first-order KBs. An important problem then is to find models of belief with better computational properties. [531]

Clearly, the purpose here is not to build a KB, nor a facility for introspective reasoning about a KB, but rather, to define a model of introspective reasoning with desirable properties—a model that may then serve as a design or specification for implementations of introspective reasoning.

In addition to defining models, papers extended, generalized, differentiated, and provided semantics for models. An example of each follows:

> **Extension**: We ... extend the notion of constraint satisfaction problems to include constraints about the variables considered in each solution. ... By expressing conditions under which variables are and are not active, standard CSP methods can be extended to make inferences about variable activity as well as their possible value assignments. [25]
>
> **Generalization**: An interesting result of our analysis is the discovery of a subtask that is at the core of generating explanations, and is also at the core of generating extensions in Reiter's default logic. Moreover, this is the subtask that accounts for the computational difficulty of both forms of reasoning. [343]
>
> **Differentiation**: While belief functions have an attractive mathematical theory and many intuitively appealing properties, there has been a constant barrage of criticism directed against them...We argue that all these problems stem from a confounding of two different views of belief functions. [112]

> **Semantics**: Their scheme has one immediate drawback; at the present moment the costs have no adequate semantics. ... we will provide a probabilistic semantics for cost-based abduction. [106]

Many papers presented theorems and proofs that derived from models. Sometimes these analyses pertained to soundness, completeness, decidability. Often they pertained to complexity; for example, [550] presents complexity analyses of eight classes of closed-world reasoning.

**Fields 5 and 6, Present algorithm(s) and Analyze algorithms.**

More than half of the papers in AAAI-90 presented algorithms, and most of these analyzed their algorithms in some manner. Complexity analyses predominated, but other kinds of formal analyses (e.g., soundness and completeness results) were common.

**Fields 7 and 8, Present system and Analyze aspect(s) of system.**

Several criteria were used to decide that a paper presents a system or an architecture. Systems and architectures are composite, with different components responsible for different functions. Systems solve problems that system designers believe are too large to be solved by a single algorithm. Frequently, system papers discussed the organization and interactions among the system's components. Although system papers often discussed just one component in detail, the discussion usually included a brief description of the system to set the context, and it was usually clear that the focal component was responsible for only some of the reasoning necessary to a task. Systems papers rarely described underlying models, theorems or algorithms. Even when they used the word "algorithm," they typically meant the flow of control in the system; for example,

> The basic QPC algorithm consists of four steps:
> 1. Assemble a view-process structure from a description of the scenario.
> 2. Apply the closed-world assumption and build the QDE.
> 3. Form an initial state.
> 4. Simulate using QSIM. [366]

Analyses of systems were divided into three classes: complexity or other formal analysis, informal, and none. As one might expect, complexity analyses focused on the behaviors of particular components of an system; for example,

> Building the space of interactions, identifying a candidate path, elaborating structure, and testing consistency are at worst quadratic in the number of individuals and classes introduced. We are working on proving whether Ibis generates all candidates; the other steps are complete. ... The verification step is NP-hard. [356]

Of the 45 papers that presented systems, seven offered complexity analyses or other formal analyses. Informal analyses included discussions of design

decisions, comparisons with related architectures, and so on. A good example is Redmond's analysis of the length of a "snippet" in his CELIA system [308].

**Fields 9, 10,11: Example type, Task type, Task environment.**

Three fields dealt with the context in which ideas were presented and tested. Most of the papers (133, or 89%) presented at least one example of a task (field 9), but only 63 of the papers (42%) indicated that their techniques had been exercised on a *task*—on multiple trials beyond a single example. Tasks were classified by type (field 10); task environments by whether they were *embedded* (field 11). Examples and task types were classified as natural, synthetic, and abstract. To be classified as performing a natural task, a program had to tackle a problem solved by humans or animals, given the same or similar data. For example,

> What we would really like to know about a function-finding program is not its record of successes on artificial problems chosen by the programmer, but its likelihood of success on a new problem generated in a prespecified environment and involving real scientific data. ... When analyzing bivariate data sets ... published in the *Physical Review* in the first quarter of this century, E* has approximately a 30 percent chance of giving the same answer as the reporting scientist. [832]

> Our system runs a complete loop in which experiments are designed by FAHRENHEIT, performed under the control of [a] PC in the electrochemical cell, [and] the experimental results are sent to ... FAHRENHEIT [which] uses them to build a theory. Human interference is reduced to sample preparation and occasional assistance. [891]

The latter excerpt describes an embedded task environment—one in which the principal innovations of a paper are applied in the context of an architecture or other software systems, or in a physical environment—whereas the former excerpt describes an algorithm, E*, that apparently runs in batch mode and has no significant interactions with its task environment. FAHRENHEIT and robot agents [e.g., 796, 854] are embedded in a physical task environment; but most embedded task environments are software environments.

Examples and tasks were classified as synthetic if they were synthetic analogs of natural tasks. For example,

> The Tileworld can be viewed as a rough abstraction of the Robot Delivery Domain, in which a mobile robot roams the halls of an office delivering messages and objects in response to human requests. We have been able to draw a fairly close correspondence between the two domains.[184]

Synthetic tasks involved simulated environments [e.g., 59, 86, 183], some planning tasks [e.g., 152, 158, 1016, 1030], distributed problem solving [78,86], and some qualitative physics tasks [e.g., 401, 407].

Whereas synthetic tasks usually raise several research issues, abstract tasks are designed to be the simplest possible manifestation of a single research issue. John Seeley Brown called such problems "paradigmatic" in his address at IJCAI 8 (1983) and distinguished them from "toy" problems, which are similarly minimalist, but are not distillations of important research issues. For example, the N-Queens problem is a paradigmatic constraint satisfaction problem, Sussman's anomaly is the paradigmatic subgoal interaction problem, ostriches and elephants provide the paradigmatic default inheritance problem, and so on. The abstract problems addressed in AAAI-90 included N-Queens [e.g., 17, 227]; constraint networks [e.g., 10, 40, 46]; networks with and without hidden variables [e.g., 556]; subgoal interaction problems [166]; problems involving multiple agents such as prisoners' dilemma, the convoy problem [94], and block-stacking problems [100] (see [538], footnotes 3 and 4 for many others); a wide variety of problems of nonmonotonic reasoning, including inheritance problems [e.g., 627, 633], qualification problems such as "potato in the tailpipe" [158], the Yale Shooting Problem [e.g., 145, 524, 615] and other problems of persistence, and various "paradoxes" of nonmonotonic reasoning [600]; a variety of simple robot problems such as the robot recharging problem [151]; problems involving artificial, large search spaces [e.g., 216]; problems involving matching [e.g., 685, 693, 701]; and a wide variety of paradigmatic classification problems for learning systems, such as XOR [e.g., 789], LED display [e.g., 762, 834], cups [e.g., 815, 861], and wins in tic-tac-toe [e.g., 803,882].

### Field 12: Assess Performance

AI is unlike experimental sciences which provide editorial guidance and university courses in experiment design and analysis. This may explain why some papers among the 160 accepted by AAAI-90 assessed performance much more convincingly than others. These papers did not set the standard for assessing performance in this survey, in part because clean experimental work is easiest when we are evaluating simple artifacts such as individual algorithms, and we didn't want to penalize efforts to evaluate complicated systems; and in part because we wanted to err conservatively, crediting too many papers with assessing performance instead of too few. Thus we adopted a weak criterion: If a paper reported a study in which at least one performance measure was assessed over a reasonably large sample of problems, then it was credited with assessing performance. "Reasonably large" is open to interpretation, but for most tasks a single example did not suffice (but see [796]). A single example usually does not explore the range of initial conditions or parameterizations that might affect performance; for instance, the following "experiment"—a single example presented without a hint of others—is inconclusive:

> Using the guaranteed planning strategy ... [the] query is solved ... in 4.75 seconds. Using the approximate planning strategy ... the same query is solvable in 0.17 seconds. Although [this] plan is not correct, it is plausible. Note also that the first two actions it prescribes are the same as those of the

> correct plan: the approximate plan is an excellent guide to intermediate action.

The "multiple examples" criterion excluded some papers in Knowledge Representation which offered a single example of a solution to, say, the Yale Shooting Problem. It is tempting to believe that a solution to such a paradigmatic problem is also a solution to countless other problems in that class. But because many of the authors of these papers did not believe this themselves [e.g., 1082] and acknowledged the need for empirical work specifically where the expressivity/tractability tradeoff was concerned [e.g., 531, 563, 633], we did not credit KR papers with assessing performance given a single example.

In general, authors did not describe the structure of their studies (just as they rarely described the purpose of studies, beyond saying the purpose was to test their ideas). We often had trouble determining the number and degree of coverage of tests in the study; but only in the most extreme cases, where authors asserted success without providing any details of the evaluation study, did we decline to credit them with assessing performance. For example:

> Our evaluation using professional and amateur designers showed that contextualized learning can be supported by [our system].

> An active critiquing strategy has been chosen and has proved to be much more effective.

### Field 13: Assess Coverage

Another purpose of an evaluation study is to assess coverage, that is, the applicability of a technique (algorithm, architecture, etc.) to a range of problems. In general, authors do not discuss their own criteria for coverage. They demonstrate techniques on problems that are superficially different, but they do not discuss whether and how they are different. Examples include an operating system and a word processing system [310]; simple liquid flow, boiling, and a spring/block oscillator [380]; the heart and a steam engine [413]. In fact, these authors do not explicitly claim coverage, so it is merely curious that they do not describe why they selected these particular suites of problems. One positive example of coverage that involves only three examples is Liu and Popplestone's paper on robotic assembly [1038]. Because they have an underlying mathematical model of their task, they were able to select examples that demonstrate coverage with respect to the model.

There are at least two alternative criteria for demonstrating coverage:

> *weak coverage:* the ability to solve instances of some problems in a space of types of problems

> *strong coverage:* the ability to solve instances of all problems in a space of types of problems

Weak and strong coverage are not distinguished in Table 1 because there were no examples of strong coverage in AAAI-90. However, one paper clearly had strong coverage as a goal [59]. It identified six "basic operations in knowledge processing": inheritance, recognition, classification, unification, probabilistic reasoning, and learning. Then it presented a knowledge processing architecture which it evaluated with problems that required inheritance, recognition, and classification.

### Field 14: Compare performance.

Although a significant number of papers included comparisons of performance, the purpose of these comparisons was not always clear. When the purpose was clearly to demonstrate that "my technique is better than yours," the paper was classified as an assessment of performance (field 12). When the purpose was to study the relative strengths and weakenesses of two or more techniques, the paper was classified as a comparison of performance. For example:

> The goal of our experiments is to draw an overall picture as to the relative strengths of back propagation and genetic algorithms for neural network training, and to evaluate the speed of convergence of both methods. ... Convergence of genetic algorithm based neural network training was so slow that it was consistently outperformed by quickprop. Varying parameters ... made only limited contributions in reversing the results [789]

### Field 15: Predictions, hypotheses

Some papers offered predictions or hypotheses; for example,

> Hypothesis 1: Only when we have a priori knowledge about problem distribution is it effective to learn macro rules. ... Hypothesis 2: As we increase the degree of nonlinearity of the recursive rules, there is exponential degradation in performance upon addition of macro rules. [947]

> The fact that filtering is less detrimental in the faster environment leads us to hypothesize that there may be a break-even point at even faster speeds, above which filtering is useful. [188]

Sometimes papers presented counterintuitive predictions. For example,

> Our account predicts (perhaps counterintuitively) that an agent will persist in trying to achieve a goal even if he happens to believe the other agent is in the process of informing him of why he had to give it up. [99]

> One might expect that in such a situation, even if the agents use the Unified Negotiation Protocol, they will agree on a semi-cooperative deal that is equivalent to the cooperative deal. ... Surprisingly, this is not the case. [104].

Hypotheses and predictions indicate that the researcher has some reason, besides demonstrating performance, to implement and test an idea. For example, the first two excerpts above hypothesize *tradeoffs* which are to be examined empirically. The third predicts (in hypothetico-deductive manner) a behavior that, though counterintuitive, follows logically from a theory; and the fourth also juxtaposes intuition and theory. Lastly, you may recall excerpts presented earlier that point out that predictions from a theory depend on the assumptions that underlie the theory, so they must be checked empirically.

These and related reasons for empirical work were sufficient to classify a paper as presenting hypotheses or predictions. What did *not* count, even when it was called a hypothesis, was the simple assertion that a technique (algorithm, architecture, etc.) solves a problem. This assertion was made implicitly or explicitly by almost all of the papers. Nor did descriptions of "experiments" imply hypotheses if they served only to demonstrate an idea (e.g., "The goal of the experiment is to make Genghis learn to walk forward." [799]). Nor did worst-case complexity results count as predictions. As noted earlier, they are technically predictions, but they predict nothing about average-case performance.

Only 25 papers contained anything that, by these criteria, could be called hypotheses or predictions. The others were vague about their reasons for empirical work. The following quotes are typical: "We implemented the above search techniques for parallel search ... and studied their performance." and, "To evaluate the effectiveness of our approach, we implemented a simulation environment and solved the [...] problem."

### Field 16: Probe results

Probing refers to a variety of activities, including explaining or strengthening experimental results (possibly with the aid of follow-up experiments), explaining results derived by other researchers, and exploratory experiments to find out more about a functional relationship thought to underlie data. In general, if a paper went beyond its central results, or explained someone else's results, it was credited with probing results. For example, the following excerpt describes how a follow-up is expected to explain the success of an earlier experiment:

> If evaluation and not search is the key to successful function-finding with real data, it ought to be possible to improve performance by developing more sophisticated evaluation criteria. [828]

And this excerpt is from a paper that develops a mathematical theory that explains why another researcher's technique works:

> Warren has proposed a heuristic for ordering the conjuncts in a query: rank the literals according to increased cost. ... It is not clear why his cost measure, and its use in this way, is appropriate. However, it becomes clear when the relation to our analysis is established. [38]

### Field 17: Present unexpected results

Very few papers discussed their results with any sense of surprise or discovery. Here are some that did:

> So far we have discovered two kinds of difficulties in building math model libraries. First, we found ourselves using ever more sophisticated qualitative models in order to provide enough functional dependencies to yield reasonable numerical models. ... [385]

> An interesting result of our analysis is the discovery of a subtask that is at the core of generating explanations, and is also at the core of generating explanations in Reiter's default logic. [343]

> These results are much better than we expected, especially when compared to ... [what] we had thought was an optimistic measure... [691]

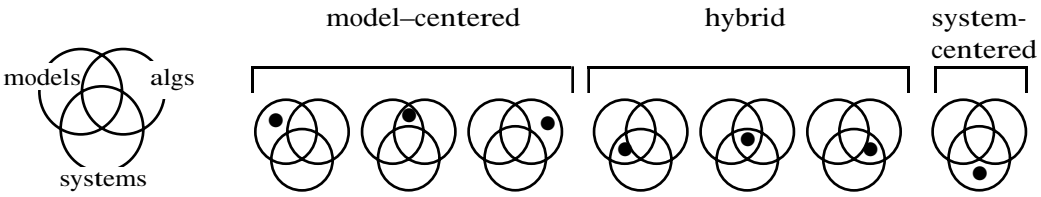
> Contrary to intuition, the random training sets performed as well or better than the most-on-point and best-case training sets. [845]

### Field 18: Present negative results

Negative results are typically things that were expected to work but did not. Examples include Hirschberg's test of an algorithm for assigning intonation to speech (discussed above), and this excerpt: "It is also probably worthwhile to report on search heuristics that we tried, but that didn't reduce the time needed to find a solution to the puzzle." [214] Evidently, most researchers were even less enthusiastic—negative results appeared in only four papers.

## Appendix 2. Statistical Analyses.

The following analyses support the conclusions in Section 4.1. Recall that papers were classified by fields 3 – 8 of Table 1 into seven sets, shown in Figure 2. These sets are shown graphically at the top of Figure 3, and the original distribution of papers into these sets is shown in row 0 of Figure 3. Consider one of these sets, say, the 43 papers in MODELS ∩ ALGS. The remaining rows in Figure 3 show how these papers are distributed over methodological tactics represented by fields 9 – 18 in Table 1. For example, the rows labelled A in Figure 3 correspond to field 9 in Table 1, which asks what kind of example was presented in a paper. The 43 papers in MODELS ∩ ALGS are distributed as follows: four papers gave natural examples, four gave synthetic examples, 29 gave abstract examples, and six gave no examples at all.




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | Distribution by fields 3 – 8 | 25 | 43 | 36 | 1 | | | |
| A | **Field 9**: Natural | 3 | 4 | 9 | 0 | | 1 | |
| A | Synthetic | 1 | 4 | 4 | 1 | | 1 | |
| A | Abstract | 15 | 29 | 20 | 0 | | | |
| A | None | 6 | 6 | 3 | 0 | 0 | 1 | 0 |
| B | **Field 10**: Natural | 0 | 3 | 10 | 0 | | 1 | |
| B | Synthetic | 0 | 0 | 2 | 1 | 1 | 2 | 3 |
| B | Abstract | 0 | 7 | 12 | 0 | | | |
| B | None | 25 | 33 | 12 | 0 | | | |
| C | **Field 11**: Embedded | 0 | 1 | 3 | 1 | | 3 | |
| C | Not Embedded | 0 | 9 | 21 | 0 | | | |
| C | None | 25 | 33 | 12 | 0 | | | |
| D | **Fields 12 –14**: Demo | 0 | 7 | 24 | 0 | | | |
| D | No Demo | 25 | 36 | 12 | 1 | | | |
| E | **Fields 15 – 18**: Expectations | 2 | 10 | 11 | 1 | | | |
| E | No Expectations | 23 | 33 | 25 | 0 | | | |

**Figure 3.** Distributions of papers by classes and fields.

Now, while the 43 papers in MODELS ∩ ALGS are predominantly concerned with abstract examples, the 37 papers in SYSTEMS (the last column in Figure 3) are concerned with natural and synthetic examples. The question is whether this result could have arisen by chance, or whether it reflects different methodological tactics. There are two ways to answer the question. One is to consider the entire distribution in the rows labelled A in Figure 3, that is, four types of example (including none) crossed with seven sets of papers. Intuitively, the distribution seems unlikely to have occured by chance; for example, we do not expect chance to distribute 15 of 25 the papers in MODELS into the "abstract" category whilst distributing 22 of 37 papers in SYSTEMS into the "natural" category. A chi-square test captures this intuition and tells us whether the entire distribution (not only the anomalous examples that we pick out) could have arisen by chance. Given the *contingency table* in the rows labelled A in Figure 3, a chi-square statistic ($\chi^2$) and its probability (p) are easily calculated. In this case, $\chi^2(18) = 67.9$ and $p < .0001$, which means that the methodological choice of an example is *not independent* of which class (e.g., MODELS, MODELS ∩ ALGS, etc.) a paper comes from; if the choice *was* independent of class, then the distribution would be expected by chance less than one time in ten thousand.

The other way to see whether the distributions in Figure 3 reflect different methodological tactics is to combine the original seven sets into three: model-centered papers, system-centered papers, and hybrid papers. As shown at the top of Figure 3, our scheme for doing this is:

- Papers in MODELS, ALGS and MODELS ∩ ALGS are model-centered (104, total)
- Papers in SYSTEMS are system-centered (37, total)
- Papers in MODELS ∩ ALGS ∩ SYSTEMS, MODELS ∩ SYSTEMS, and ALGS ∩ SYSTEMS, and are hybrid (eight, total).

One justification for this is that the papers we call model-centered cannot be differentiated by the kinds of examples they contain. To see this, we construct a contingency table from the first three columns of data in the rows labelled A in Figure 3. This distribution, shown in Figure 4, does not permit us to reject the hypothesis that example type is statistically indendent of the classification of a paper as a member of MODELS, ALGS or MODELS ∩ ALGS ($\chi^2(6) = 7.26$, $p > .29$).

| | Field 9: | | | |
|---|---|---|---|---|
| A | Natural | 3 | 4 | 9 |
| A | Synthetic | 1 | 4 | 4 |
| A | Abstract | 15 | 29 | 20 |
| A | None | 6 | 6 | 3 |

**Figure 4.** Three classes of papers with the same distribution of types of examples.

With this as a justification for the class of model-centered papers, the other classes follow naturally: system-centered papers are those in SYSTEMS, and the

remaining eight, hybrid papers are those in the intersection of the model-centered and system-centered classes.

Now we can run chi-square tests as above, except with three classes instead of seven. New contingency tables are easily derived by summing over columns; for example, Figure 5 shows the new table for the rows labelled A in Figure 3. This distribution is unlikely to have arisen by chance ($\chi^2(6) = 55.5$, $p < .0001$), which means that model-centered and system-centered papers offered significantly different types of examples.

| | Field 9: | Model-centered | Hybrid | System-centered |
|---|---|---|---|---|
| A | Natural | 16 | 1 | 22 |
| A | Synthetic | 9 | 4 | 11 |
| A | Abstract | 64 | 2 | 4 |
| A | None | 15 | 1 | 0 |

**Figure 5.** The contingency table derived from Figure 3 for the distribution of example types over three classes of papers.

Exactly the same sort of analyses were run on fields 10 and 11, with the results reported in Section 4.1: model-centered and system-centered papers focussed on significantly different kinds of tasks and task environments. The data are shown in the rows labelled B and C, respectively, in Figure 3. Note, however, that to analyze the embedded/non embedded distinction, we constructed a contingency table that included only papers which described a task (in fact, we left out the row labelled "C. None" in Fig. 3) because it makes no sense to ask whether a task environment is embedded if there isn't a task.

Three analyses warrant further explanation. First, we had to combine data from fields 15 – 18 into a single "super field" called *expectations* (a "yes" in at least one of the fields counted as an expectation). The rows labelled E in Figure 3 show the distribution of expectations. The contingency table for model-centered, system-centered, and hybrid papers was derived as described above, and showed that model-centered and hybrid papers were more likely than system-centered papers to discuss expectations ($\chi^2(2) = 10.5$, $p < .01$).

Second, we combined the data in fields 12 – 14 as shown in the rows labelled D in Figure 3. These show the distribution of *demonstrations* over all papers. But we also ran an analysis of the distribution of demonstrations over *papers that described tasks* (field 10, see also rows B in Fig. 3). The contingency table in Figure 6 shows that among the papers that described a task, model-centered papers were more likely than system-centered papers to present a demonstration ($\chi^2(2) = 19.97$, $p < .001$).

| Fields 12 - 14 Papers with tasks: | Model-centered | Hybrid | System-centered |
|---|---|---|---|
| Demo | 31 | 5 | 8 |
| No Demo | 3 | 1 | |

**Figure 6.** The contingency table for the distribution of demonstrations in papers that described tasks, over three classes of papers.

Finally, to test whether model-centered or system-centered papers analyzed their results to different extents, we had to change slightly the definitions of these classes. Recall that MODELS papers are those that presented models (field 3) *or* proved theorems about the models (field 4). Let us change the definition of MODELS to include those papers that garnered a "yes" in field 3, only, and count a "yes" in field 4 as evidence of analysis of models. Similarly, let a "yes" in field 5 or 7 assign a paper to ALGS or SYSTEMS, respectively, and a response in field 6 or 8 count as evidence of analyzing the algorithm or system, respectively. Then the definitions of model-centered, hybrid, and system-centered are as they were before, and the contingency table relating these classifications to the distribution of analyses is shown in Figure 7. Clearly, model-centered papers and hybrid papers (as redefined) are more likely than system-centered papers to present analyses ($\chi^2(2) = 16.5$, $p < .0005$).

| Fields 4, 6, *or* 8 : | Model-centered | Hybrid | System-centered |
|---|---|---|---|
| Analysis | 82 | 6 | 16 |
| No Analysis | 22 | 2 | |

**Figure 7.** The contingency table for the distribution of analyses over three classes of papers.

## Problems with and Concerns about the Survey.

It would be misleading to end this discussion without addressing some problems with our own methodology—the way we conducted the survey. The major problem is that we have no reliability data. We cannot be confident that another reviewer, given the fields in Table 1, would classify the papers in substantially the same way. To illustrate the problem, consider the most difficult question we had to tackle in the current survey: where to draw the line between "informal analysis" and "no analysis" of systems (field 8). The line must distinguish real analyses from guesses, post-hoc justifications, wish-lists for extensions to the system, perfunctory and obligatory references to other research, and so on. The criteria for this distinction are very subjective; however, we needed some way to acknowledge the 13 papers that in an ill-defined way tried to analyze their systems (especially because nine of them had not a single non-negative entry in fields 12 – 18). We believe that other questions in Table 1 can be answered more

objectively, but to find out will require a reliability study, for which we solicit volunteers!

Bias due to preconceptions is another concern. Perhaps by identifying a paper as, say, a member of SYSTEMS, we became biased in how we filled in the other fields in Table 1. For example, we might be more likely to classify an experiment as a demonstration of performance if it came from a SYSTEMS paper than an ALGS paper because we *expected* SYSTEMS papers to demonstrate performance more often than ALGS papers. In fact, we expected exactly this but we found the opposite, so the bias—if it existed—was clearly not strong enough to eradicate the true result in this instance. Bias is probably a factor in our results, but we doubt it is a major factor—or at least we have not discovered obvious examples of it.

We must also address the concern that the papers in AAAI-90 do not represent the methodological status of AI. Perhaps methodologically superb work is being excluded by space limits, reviewing criteria, or other factors in the AAAI reviewing process. We found no evidence to suggest that the reviewing process is a Maxwell's Demon that lets bad work in and keeps good work out. Roughly 80% of AAAI-90 papers provided either analysis or demonstrations of performance, which suggests that the program committee was looking for *something* to back up the claims made in the papers; and the fact that roughly 20% provided neither analysis nor demonstrations suggests not that superb work was rejected, but that it was hard to come by. Perhaps, then, it is not the reviewing process but the requirements of the forum itself—particularly the page limits—combined with self-selection, that prevent researchers from sending their best work to AAAI. No doubt there is something to this, but it is not the simplest explanation of the wide variance in the quality of work in AAAI-90.