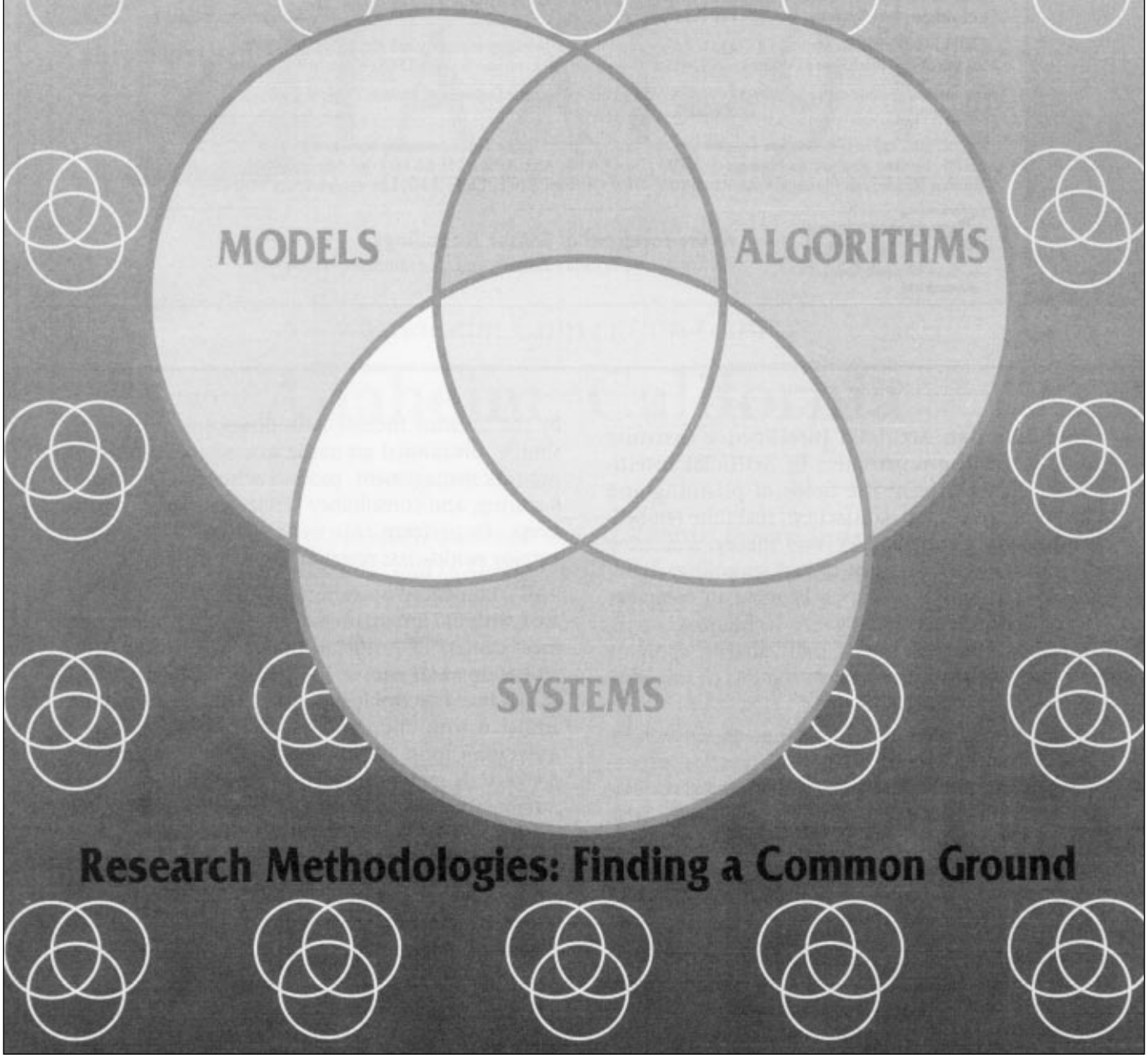
A Survey of the Eighth National Conference on Artificial Intelligence: Pulling Together or Pulling Apart?

Paul R. Cohen



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.

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, and perception and underpinnings of these functions such as commonsense knowledge and reasoning. We could debate whether it makes sense to study intelligence piecewise—you solve vision and I solve planning, and someday we might get together to build autonomous mobile robots—but this concern is not the main one 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, I present the results of a survey of 150 papers from the Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90) (AAAI 1990). I 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 a lack of evaluation, a lack of hypotheses and predictions, irrelevant models, and weak analytic tools. Fourth, a methodology exists that merges the current "big two" and eliminates the conditions that give rise to methodological problems. My survey provides direct statistical support for the first claim; the other claims are supported by statistical evidence and excerpts from the papers presented at AAAI-90.

This presentation has three parts: a summary of the survey and general results, a discussion of the four hypotheses, and two sections at the end of the article that contain details of the survey and statistical analyses. The next section (The Survey) briefly describes the 16 substantive questions I asked about each paper. One of the closing sections (An Explanation of the Fields in Table 1) discusses the criteria for answering the survey questions and illustrates the criteria with excerpts from AAAI-90 papers. In General Results, broad, descriptive statistics characterize the papers, whereas statistical tests of my hypotheses are described in Four Hypotheses and in Statistical Analyses. Arguments against my proposed methodology (introduced in Hypothesis 4: A Sufficient Methodology Exists) are considered but not conceded in Anticipating Arguments against Modeling, Analysis, and Design.

I acknowledge that methodological papers are unpalatable for a variety of reasons. However, they indicate that the field is approaching maturity (DeMey 1982) and, thus, should be welcomed for this reason if not for the problems they raise. In fact, this article is extremely positive because unless I badly misread the field, it should be easy to remove the structural, endogenous causes of our methodological problems. Then, we only have to worry about conservatism and other sociological impediments, which can be addressed in curricula and editorial policy.

Not only were average-case hypotheses and predictions rare, so too were follow-up experiments.

The Survey

The survey covered 150 of the 160 papers from AAAI-90. I read all the papers and omitted the 10 that I did not understand or that did not easily fit into table 1. Each paper is characterized by the 19 fields in table 1. I only briefly describe these fields here to quickly get to the survey results (the reader should consult An Explanation of the Fields in Table 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 was 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 or architectures (field 7), and analyze them (field 8). These purposes are not mutually exclusive; for example, many papers that present models also prove theorems about the models.

0738-4602/91/$4.00 ©1991 AAAI

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 embeded 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. The Classification Scheme for AAAI-90 Papers.

The number of papers in each classification is shown in the columns. For example, of 61 papers that analyze algorithms, 27 offer complexity analyses, 19 present other formal analyses, and 15 give 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 probe 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.

Models are formal characterizations of behaviors (for example, two papers from AAAI-90 present models of cooperative problem solving) or task environments (for example, several papers focus on recursive problem space structures). Some papers extend models to incorporate new behaviors (for example, extending ordinary constraint-satisfaction problem solving to include dynamic constraints on variables). Some papers generalize models, and others differentiate 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 provide formal semantics for models that previously included vague terms (for example, probabilistic semantics for costs). More than half the papers in the proceedings present algorithms (field 5), and many also analyze the algorithms (field 6). Complexity analyses dominate. Surprisingly, only 45 papers present systems (field 7), and even fewer analyze systems (field 8). The distinctions between models, algorithms, and systems are somewhat subjective and are illustrated in An Explanation of the Fields in Table 1.

Fields 9–18 in table 1 represent methodological tactics for convincing the reader that the purpose of a paper was achieved. The

most common tactic was to present a single example (field 9), but many papers report studies involving multiple trials designed to assess performance (field 12), 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, and 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 (for example, simulations of robots in dynamic environments); abstract examples and tasks are designed to illustrate a single research issue in the simplest possible framework (for example, *N* queens, the Yale shooting problem, Sussman's anomaly). Some papers describe techniques embedded in a larger environment (for example, temporal projection embedded in a planning system).

Relatively few papers present hypotheses or predictions (field 15). The criteria for what counts as hypotheses and predictions are discussed in An Explanation of the Fields in Table 1. However, because the absence of hypotheses in AAAI-90 papers is central to this article, I must note here that worst-case complexity results—which are 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 or how techniques will behave in average cases. Not only are average-case hypotheses and predictions rare, but follow-up experiments to probe previous results (field 16) and reports of negative and unexpected results (fields 17 and 18) are as well. Because hypothesis testing, follow-up studies, and replications with extensions are common and compelling methodological tactics throughout the sciences, their absence from AAAI-90 papers is troubling.

The survey involved subjective judgments, but no reliability studies were performed. This caveat and related concerns are discussed further in Statistical Analyses. 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 An Explanation of the Fields in Table 1. Excerpts from the papers are referenced by the following convention: Each is identified by a single number that is either the page number in the proceedings on which the excerpt can be found or the page number of the first page of the paper. A few excerpts are unattributed.

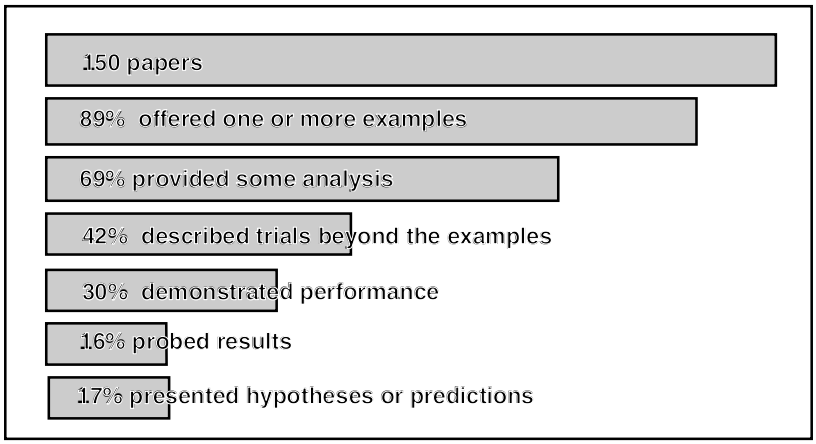


Figure 1. Summary of Results from the Survey of Papers in AAAI-90.

General Results

Of the 150 papers surveyed, most include one or more examples (field 9), but fewer than half describe a task and trials of a system beyond a single example (field 10); only 45 papers demonstrate performance in some manner (fields 12, 13, and 14). One hundred and four papers offer some kind of analysis (see definition later). Twenty-four papers probe or otherwise examine results (fields 16, 17, and 18), and 25 papers present hypotheses or predictions (field 15).

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

Four Hypotheses

"AI is two schools of thought swimming upstream."

—*C. R. Beal*

This survey provides support for four hypotheses about the current state of AI research: First, most AI research is conducted with two methodologies that in the past have 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 for its goals. Fourth, by combining aspects of the two methodologies, we get

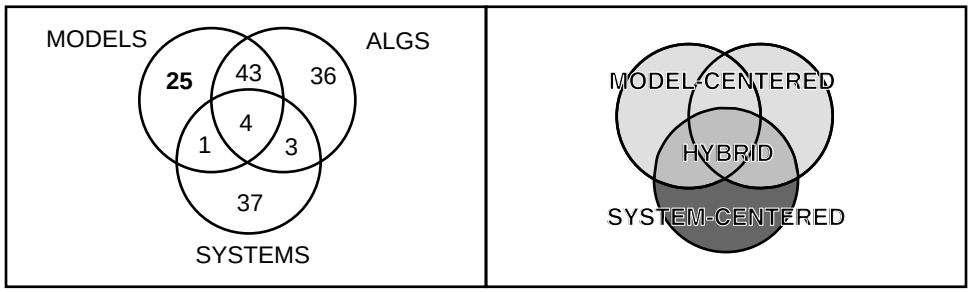


Figure 2. Papers in AAAI-90 Classified by Fields 3–8.
A. By number. B. By methodology.

another methodology that is less prone to these methodological problems. The following subsections discuss the evidence for these hypotheses. Hypothesis 1: Two Methodologies presents statistical evidence to support the two-methodology hypothesis; the other hypotheses are supported by a combination of statistical evidence and excerpts from the AAAI-90 papers. The third hypothesis, which claims a causal relationship, is only indirectly supported when I show that methodological problems are present when the two methodologies are individually practiced and absent when aspects of the two methodologies are combined. The third hypothesis is important because it claims that many or all of AI's methodological problems have a common root, thus suggesting that these problems can be corrected en masse. The fourth hypothesis is supported by descriptions and demonstrations of aspects of the combined methodology presented in some AAAI-90 papers.

Hypothesis 1: Two Methodologies

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

I used fields 3–8 of table 1 to classify the papers into three sets and their intersections, shown in figure 2a. The first set, MODELS, includes those papers that had "yes" in fields 3 or 4, that is, papers that deal with models. Twenty-five papers deal with models alone, 43 deal with models and algorithms, 1 deals with models and systems, and 4 deal with all 3 topics. The second set, ALGS, includes all papers that present algorithms (field 5) or some kind of analysis of the algorithms (field 6). The third set, SYSTEMS, contains papers that present systems or analyses of systems (fields 7 and 8, respectively). One paper belongs to none of these classes. This result causes some totals in the subsequent analyses to be one less than indicated in table 1.

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

Model-centered papers represent one methodology and system-centered papers another. To show that these methodologies are both real and significantly different, I adopt the following strategy: Starting with the classification of papers in figure 2b, I 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 occurred 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 Statistical Analyses, 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 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 Statistical Analyses.

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

The classes MODELS, ALGS, and MODELS ∩ ALGS (figure 2) could not be differentiated by the kinds of examples they contain. Eighty-four percent of the papers in MODELS, 81 percent of the papers in MODELS ∩ ALGS,

and 64 percent of the papers in ALGS give abstract examples or no examples at all. Because the papers in MODELS, ALGS, and MODELS ∩ ALGS present the same kinds of examples with roughly the same relative frequencies, one is justified in combining the papers in these sets into the single class of model-centered papers. The papers in MODELS are *preimplementation* 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.

Recall that some papers describe *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: Eighty-five percent of the model-centered papers describe abstract tasks or no tasks at all, whereas 58 percent of the 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 addresses a task, which is not surprising if we view them as preimplementation papers, but I was surprised to find that 41 percent of the system-centered papers describe no task, that is, nothing more than a single illustrative example. Still, a greater proportion of system-centered papers (59 percent) than model-centered papers (33 percent) describe tasks ($\chi^2(2) = 11.9, p < .005$).

Of the papers that do report multiple trials on tasks, 86 percent of the system-centered and hybrid papers describe embedded task environments, whereas 88 percent of the model-centered papers describe 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 real physical environments.

Model-centered and system-centered papers differ in their orientation toward assessing performance, assessing coverage, and comparing performance (fields 12, 13, and 14, respectively). A paper presents a *demonstration* if it reports at least one of these three activities. Remarkably, a higher proportion of model-centered papers (30 percent) than system-centered papers (22 percent) present demonstrations, even though 25 model-centered papers (from MODELS) are preimplementation 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 describe 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: Thirty-six percent of the system-centered papers that describe a task also present demonstrations, compared with 91 percent 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 describe multiple trials on tasks, relative to model-centered papers, fewer present 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 An Explanation of the Fields in Table 1 for a list of abstract problems.)

...system-centered papers represent different methodological tactics than model-centered papers.

Recognizing that demonstrations are only one way to evaluate a technique (and not a particularly informative one at that), I looked at whether system-centered and model-centered papers have different propensities to analyze their contributions. I found that 79 percent of the model-centered papers, 75 percent of the hybrid papers, and just 43 percent of the system-centered papers report any kind of analysis. This result 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 Statistical Analyses).

Finally, I looked at the relative frequencies of hypotheses, predictions, probes, unexpected results, and negative results (fields 15–18, respectively). I had hoped to analyze these fields separately, but only field 15 (hypotheses, predictions) contained enough data to support statistical tests. By combining the fields, I was asking whether the researcher had any expectations beyond the common assertion that a technique will work (see An Explanation of the Fields in Table 1 for descriptions of fields 15–18). Once again, I found a significant effect of methodology: Twenty-two percent of the model-centered, 11 percent of the system-centered, and 62.5 percent of the hybrid papers have expectations ($\chi^2(2) = 10.5, p < .01$). Although few papers overall give evidence of expectations, the model-centered and hybrid papers do so more often than the system-centered papers, suggesting that the models in the model-centered papers might offer a small advantage in

…some models are inadequate for predicting and analyzing… behavior… others are not used in this way.

generating hypotheses and predictions.

The paucity of expectations in fields 15–18 is disturbing; so, I asked whether evidence of expectations could be found in other fields in table 1. One possibility is field 14, which I used to register papers that compare performance among techniques. I reasoned that the techniques were not arbitrarily selected; they were meant to probe or explore expectations about their relative strengths and weaknesses. Remarkably, model-centered papers number 20 of the 24 that compare 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, I 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. Model-centered researchers, however, typically develop algorithms for simple, abstract problems but with deeper analysis and expectations and more demonstrations of success.

Hypotheses 2 and 3: Insufficient Methodologies Cause Methodological Problems

I am developing a case that comprises four claims: There are two AI methodologies; alone, neither is sufficient; almost nobody is using both methodologies together; and, in combination, the methodologies are sufficient. My results are unequivocal for the first and third claims: The two methodologies are real enough, involving different methodological choices, and only 8 of 150 papers bridged the methodologies. This subsection presents evidence that the methodologies are not sufficient, and the next subsection argues that a composite methodology is sufficient. Along the way, I 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, and so on—and one immediately sees that central to design is the ability to predict and analyze the behavior of the systems. However, my survey shows virtually no interaction between researchers who develop models that are in principle predictive and analytic and researchers 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, and others are not being used in this way.

I 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. In their AAAI-90 paper, Tambe and Rosenbloom rely on two kinds of models to discuss issues in the design of production match algorithms. First, 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 requires 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 the AAAI-90 proceedings (for example, 633, 640) and elsewhere, that this reduction in complexity is bought at the cost of expressiveness, the so-called expressiveness-tractability trade-off (Levesque and Brachman 1985).

Trade-offs are essential to designers because they can be used to predict—if only comparatively and qualitatively—the behavior of alternative designs. The most useful trade-offs are operational in the sense of telling the designer what to change—which knobs to tweak—to change behavior. The expressiveness-tractability trade-off 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 are not concerned with systems (that is, they lack architectural knobs to tweak), they do not operationalize the expressiveness-tractability trade-off (or other trade-offs). In addition, 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 trade-off, but until we build a system, there is 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 trade-off by exploring it in the context of production system architectures. This approach 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 (for example, restrictions on the number of values for each attribute). They show that within this framework, the unique-attribute 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, the unique-attribute formulation is not guaranteed to be better than other possible formulations.

Like Tambe and Rosenbloom, Subramanian and Feldman develop a model to represent a design trade-off 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 learned 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 the AAAI-90 proceedings; only 8 of 150 papers reside in the intersection of model-centered and system-centered research. Is this situation bad? I offered a couple of examples in which the methodologies are profitably merged; now I document the costs of exclusively working in one methodology. This demonstration is most convincing when the researchers within each methodology speak for themselves. I begin with model-centered research.

Models without Systems. One concern is that the analytic tools of model-centered research do not cut finely enough, so empirical research is necessary. Worst-case complexity analysis—the most common kind of analysis presented in the AAAI-90 papers—does not tell us how systems will perform in practice. Model-centered researchers acknowledge that intractable tasks might in fact be possible, and approximations or otherwise weakened models might 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

…the analytic tools of model-centered research do not cut finely enough… empirical research is necessary.

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 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 one agrees that an abstract problem is representative of a natural one, solving the former might 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 might 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 might not arise in practice, lose track of the purpose of a kind of reasoning, not exploit practical constraints when designing solutions to formal problems, and 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 when a researcher attempts to use models in systems, which is extremely rare in the AAAI-90 papers. However, virtually all model-centered papers are prone to these problems. Consider that 76 percent of the model-centered papers give abstract examples or no examples; only 33 percent of these papers describe tested implementations, and more than half of these implementations are tested on abstract problems; only 4 model-centered papers describe techniques embedded in larger software or hardware environments.

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, and design. 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.

> 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, I turned to Lenat and Feigenbaum's (1987) 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. (p. 1177)

Apparently, the methodology is not being practiced by system-centered researchers or is not producing the desired results. The 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 present hypotheses that could surprise the experimenter, and only 2 of these are system centered (the rest present the hypotheses that a program works or works better than another program or present no hypothesis at all). In addition, if nature is so full of surprises, why do only 24 papers report negative results or unexpected results or probe results?

One is tempted to criticize these papers, as Lenat and Feigenbaum (1987) 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" (p. 1177). In other words, the empirical inquiry hypothesis is okay, but AI researchers are not. However, I 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,

Only 25 papers presented hypotheses that could surprise the experimenter and only two of these were system-centered.

mathematical models, but one also cannot proceed in the 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.

I 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 claim 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 $S \rightarrow B$) are sometimes called existence proofs, nobody ever claimed that these programs could not exist; no hypothesis or conjecture is being tested by implementing them. $S \rightarrow B$ is not itself a hypothesis. Neither $S \rightarrow B$ nor its negation is 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." Few systems merit attention on the basis of their existence alone.

Second, desired behaviors are loosely specified (for example, real-time problem solving, graceful degradation, situated action), so $S \rightarrow B$ 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 (1981) in 1976, continues unabated today. Behaviors are what are produced by the components of systems that carry the names of behaviors (for example, scheduling is what the scheduler does). This transference is exhilarating—we can build anything we can imagine and call it anything we like. The downside is that what we *can* build displaces 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, one 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 (Cohen 1989; Cohen and Howe 1990, 1988a, 1988b; Langley and Drummond 1990). 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 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. With few exceptions, described later, system-centered papers in the AAAI-90 proceedings lack explicit, predictive models. Given this fact, 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. In addition, 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."

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/EBLs 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 tested the hypothesis that the filtering mechanism improves performance. Unlike most experiments presented in the AAAI-90 papers, Pollack and Ringuette's carefully varied the experimental conditions and, consequently, revealed a trade-off between the conditions (in this case, the rate of change in the environment) and performance. This result 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 the work of Minton, Johnston, Philips, and Laird, or they can evolve over a series of experiments such as Pollack and Ringuette's. The important point 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?

Hypothesis 4: A Sufficient Methodology Exists

Here, I document the evidence in the AAAI-90 proceedings of a methodology sufficient to the goals of providing science and technology to support the design and analysis of AI systems. I call the methodology *modeling, analysis*, and *design* (MAD). MAD involves seven activities: (1) assessing environmental factors that affect behavior; (2) modeling the causal relationships between a system's design, its environment, and its behavior; (3) designing or redesigning a system (or part of a system); (4) predicting how the system will behave; (5) running experiments to test the predictions; (6) explaining unexpected results and modifying models and system design; and (7) generalizing the models to classes of systems, environments, and behaviors.

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

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 among the AAAI-90 papers. 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

We can claim understanding when we can predict… how changes in design or… conditions will affect behavior.

"when designing our current media coordinator [we] showed that people strongly prefer sentence breaks that are correlated with picture breaks" (442). Many papers focus on a single aspect of environments. Time (for example, 132, 158) and the recursive structure of problem spaces (for example, 916, 336, 942) are examples. Only one paper explicitly intended to study the influences on 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. For the purposes of designing and analyzing systems, these assumptions say that it probably won't hurt to simplify the characterization of the environment. Assumptions are plentiful in the AAAI-90 papers, especially in the model-centered papers, but they are assumptions about no particular environment and, I sometimes suspected, about no plausible environment. This point 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 situation 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).

Modeling. 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-attribute 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. Minton, Johnston, Philips, and Laird give the following probability that the min-conflicts heuristic will make a mistake assigning a value to a variable:

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

The important point 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, Johnston, Philips, and Laird 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; for example, Etzioni (916) and Subramanian and Feldman (942) focus on the recursive structure of problem spaces to predict and explain problem-solving behavior. Unfortunately, many models in the AAAI-90 papers give only qualitative, worst-case characterizations of search spaces (that is, intractability results) that could not be used to answer either of the two questions. I did not classify the kinds of models developed in the AAAI-90 papers, but the paucity of hypotheses and predictions among them suggests that the models either were for some reason not being used to answer questions 1 and 2 or, more likely, were not intended to answer the questions. It seems likely that most of the models described in the AAAI-90 papers cannot support most MAD activities.

Design and Redesign. Designs, or rather sketches of designs, abound in AAAI-90 papers, especially in the 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, then designs from the models. Dechter, for example, 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 approach 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. 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 environmental change. They knew that as the run time of tasks increased, so would the benefit of filtering these tasks and, they assumed, so would the accuracy of the results. On the basis of this qualitative model, they proposed changing their design "to implement more accurate (and costly) deliberation mechanisms in the near future. For these mechanisms, filtering might be much more valuable" (188). This paper is one of a small handful in the AAAI-90 collection that justify design revisions based on models; another excellent example is de Kleer's revisions to the design of truth maintenance systems to exploit locality in the underlying structure of some problems (264).

Prediction. Prediction is central to the MAD methodology: During the design process, you predict how a system will behave; 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 a 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 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 Anticipating Arguments against Modeling, Analysis, and Design).

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 papers, few experiments served these purposes; instead, they demonstrated performance. Although demonstrations contribute little to our understanding of our systems, if we are going to keep building them, we should at least develop meaningful, efficient measures of performance. Not surprisingly, this effort can also profitably be 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 directly comparing performance, we can compare leafiness. Most performance measures in the AAAI-90 papers are not so carefully justified. Eskey and Zweben point out that a common performance measure—run-time speedup—is not a proxy for the measure that represents their goals as designers, so it should not be adopted without careful consideration (908). The correlation between run-time speedup 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

…we shouldn't think that instituting benchmarks will fix AI's methodological problems, particularly the lack of predictions and hypotheses.

effect—in borderline situations—if an experiment is 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. However, 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 contradict predictions, but often they 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 (1990), 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 search 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. (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, the Defense Advanced Research Projects Agency recently sponsored a workshop on benchmarks and metrics and is instituting benchmarks in some of its research programs. I believe that benchmarks and common environments address a symptom—the lack of system evaluation—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 performance demonstrations (Cohen and Howe 1990). I state the case too strongly—benchmarks can certainly provide common evaluation criteria and might 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.

We also shouldn't 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 (that is, the same software testbed) then the results will be comparable, replicable, and cumulative.[2] Like the call for benchmarks, this idea isn't bad, but it diverts attention from a real methodological problem. Replication in other fields is not the replication of laboratories but the replication of results across laboratories. The strongest results are those 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, so for the sake of comparability and cumulativeness 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 bench-

marks; we can compare techniques within a common experimental environment. All these techniques are valuable exploratory methods. All are preferable to unsubstantiated claims of success. However, none is half as convincing as a test of a prediction derived from a model and replicated across environments.

Explanation. By *explanation*, I mean accounting for data; for example, Minton, Johnston, Philips, and Laird account for the performance of the min-conflicts heuristic with the model previously described. However, one might also 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). In Hirschberg's case and the natural sciences in general, explanations of 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?

This question recently arose in our PHOENIX system (Cohen 1991; Cohen et al. 1989; Howe, Hart, and Cohen 1990). On the basis of a model, it was predicted that problems would most efficiently be solved in a particular order; however, the prediction failed: Performance was inefficient in one of four experimental conditions. The model included terms that represented the problem-solving environment, and it made some assumptions about the problem-solving architecture. To explain the results, we first showed that the model correctly characterized the environment and then attributed the failed prediction to one of these assumptions. This approach 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 can generate dozens or hundreds of imitations, reimplementations, and improvements. Eventually, someone writes a paper that states generally and more or less formally what all these systems do, for example, Clancey's (1985) heuristic classification paper, Mitchell's (1981) characterization of generalization as search, and Korf's (1987) paper on planning as search. The trouble is that such papers are rare.

The activities just discussed can be combined to yield several styles of AI research. I mentioned hypothesis testing, where predictions are generated from models and empirically tested in systems. I also mentioned exploratory model development, where empirical work is intended to first suggest and then refine models (Langley and Drummond 1990). Sometimes, the 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 the 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 [Cohen et al. 1989]. Instead, researchers differentiated *fixed* design decisions, which will not be reviewed anytime soon; *reviewable* decisions, which are reviewed after they are implemented, and 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 iteratively develop models, make predictions, review design decisions in light of new models, propose new design decisions, and explain performance. To date, most of the modeling effort has been aimed at analyzing reviewable design decisions; for example, although PHOENIX agents currently work on multiple fires simultaneously, we recently developed a model that suggests this approach is not the best use of resources. If the model holds up empirically, then we will revise the design decision. In sum, although

…although the MAD activities get "mixed and matched" the constant theme is… models to support the analysis and design of systems.

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.

Anticipating Arguments Against Modeling, Analysis, and Design

Here, I consider five arguments against the MAD methodology and, more generally, against any attempt to base the design and analysis of AI systems in models. I do not believe these arguments; I 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 (for example, Langton [1989]). A related argument is that AI systems are too complex to be modeled in their entirety. In fact, complex systems can be modeled, and behavior can be predicted, if not accurately, at least accurately enough to inform design. Particularly useful, as I noted earlier, are models of design trade-offs. These models need not be accurate, they might only be qualitative, but they help designers navigate the space of designs. Moreover, once a prototype design is implemented, even qualitative design trade-offs can quickly be enhanced by empirical data. It is also not necessary to model an entire system to predict its performance. By modeling a critical component of a system—a bottleneck, perhaps—one can predict the gross behavior of an entire system. Thus, the question is not whether predicting behavior is possible in principle or 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 is, 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 I cannot address here for want of space and expertise.[3] However, I can say that most of what AI seems to mean by creativity involves relatively small variations on a theme; new themes are infrequently introduced. Nothing in MAD precludes designing a system that is predicted to produce novel variations on a theme. No individual variation would be predictable, but the system would not stray from the theme.

"Premature Mathematization Keeps Nature's Surprises Hidden"

Another possible argument against MAD is that modeling discourages exploration or, as Lenat and Feigenbaum (1987) put it, "Premature mathematization keeps Nature's surprises hidden" (p. 1177). I know of no area of inquiry that has been retarded by efforts to build formal models of nature, but obviously, one's 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 system design and analysis. Models serve design and analysis. The methodology simply does not endorse modeling 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 behaviors that the system-centered researchers have assumed all along. Probabilistic accounts of certainty factors came along a decade after MYCIN (Heckerman 1986); semantics for STRIPS operators were developed later yet (Lifschitz 1987). The synchronization problem is that system-centered researchers don't get models when they need them, which is during the design and analysis of systems. I believe the problem is real, but I 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 say that systems are premature, lacking formal models of intelligence. System-centered researchers say models are superfluous because the goals of AI are satisfied if we can build systems that work, which 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 goals should be the activities of different cadres of researchers, as they are now, or whether the activities should somehow be merged. 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 simultaneously achieve them.

An Explanation of the Fields in Table 1

Fields 3 and 4: Define, Extend, Generalize, Differentiate Semantics for Models and Theorems and Proofs for the Model

Many papers focus on models of reasoning. The word model is used many ways in AI, but I intend it to mean an abstract, typically formal description of behavior 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 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 knowledge base or a facility for introspective reasoning about a knowledge base but, rather, to define a model of introspective reasoning with desirable properties, a model that can then serve as a design or specification for implementations of introspective reasoning.

In addition to defining models, papers extend, generalize, differentiate, and provide 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 present theorems and proofs that derived from models. Sometimes these analyses pertained to soundness, completeness, and decidability. Often, they pertained to complexity; for example, one paper (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 the AAAI-90 proceedings present algorithms, and most of these analyze their algorithms in some manner. Complexity analyses predominate, but other kinds of formal analyses (for example, soundness and completeness results) are 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 discuss the organization and interactions among the system's components. Although system papers often discuss just one component in detail, the discussion usually includes 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. System papers rarely describe underlying models, theorems, or algorithms. Even when they use the word algorithm, they typically mean 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 a system; for example:

Building the space of interactions, identifying a andidate 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 present systems, 7 offer complexity analyses or other formal analyses. Informal analyses include 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, and 11: Example Type, Task Type, Task Environment

Three fields dealt with the context in which ideas are presented and tested. Most of the papers (133, or 89 percent) present at least 1 example of a task (field 9), but only 63 of the papers (42 percent) indicate 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. 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 (for example, 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 involve simulated environments (for example, 59, 86, 183), some planning tasks (for example, 152, 158, 1016, 1030), distributed problem solving (78, 86), and some qualitative physics tasks (for example, 401, 407).

Synthetic tasks usually raise several research issues, but 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 the Eighth International Joint Conference on Artificial Intelligence in 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 the AAAI-90 papers include *N* queens (for example, 17, 227); constraint networks (for example, 10, 40, 46); networks with and without hidden variables (for example, 556); subgoal-interaction problems (166); problems involving multiple agents, such as the prisoner's dilemma, the convoy problem (94), and block-stacking problems (100) (see (538) for many others); a wide variety of problems of nonmonotonic reasoning, including inheritance problems (for example, 627, 633); qualification problems such as the potato in the tailpipe (158), the Yale shooting problem (for example, 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 (for example, 216); problems involving matching (for example, 685, 693, 701); and a wide variety of paradigmatic classification problems for learning systems, such as XOR (for example, 789), LED display (for example, 762, 834), cups (for example, 815, 861), and wins in tic-tac-toe (for example, 803, 882).

Field 12: Assess Performance

AI is unlike experimental sciences that provide editorial guidance and university courses in experiment design and analysis. This statement might explain why some papers among the 160 accepted for AAAI-90 assess performance much more convincingly than others. These papers are not the standard for assessing performance in this survey, in part because clean experimental work is easiest when evaluating simple artifacts such as individual algorithms, and I didn't want to penalize efforts to evaluate complicated systems, and in part because I wanted to conservatively err, crediting too many papers with assessing performance instead of too few. Thus, I 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 (however, see (796)). A single example usually does not explore the range of initial conditions or parameterizations that might affect performance; for example, the following so-called 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 example criterion excluded some papers in knowledge representation that offer 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 this class. However, because many of the authors of these papers do not believe this statement themselves (for example, 1082) and acknowledge the need for empirical work specifically where the expressivity-tractability trade-off is concerned (for example, 531, 563, 633), I did not credit knowledge representation papers with assessing performance given a single example.

In general, authors do not describe the structure of their studies (just as they rarely describe the purpose of studies beyond saying the purpose is to test their ideas). I often had trouble determining the number and degree of the coverage of tests in the study, but only in the most extreme cases, where authors assert success without providing any details of the evaluation study, did I 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, and so on) 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); and 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: (1) *weak coverage*, the ability to solve instances of some problems in a space of types of problems, and (2) *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 are no examples of strong coverage among the AAAI-90 papers. However, one paper clearly has strong coverage as a goal (59). It identifies six basic operations in knowledge processing: inheritance, recognition, classification, unification, probabilistic reasoning, and learning. Then, it presents a knowledge processing architecture that it evaluates with problems that require 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 weaknesses 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 offer 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 present 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 in this subsection hypothesize tradeoffs that are to be empirically examined. The third excerpt predicts (in a hypothetico-deductive manner) a behavior that, although counterintuitive, follows logically from a theory; the fourth also juxtaposes intuition and theory. Last, you might recall excerpts presented earlier that point out that predictions from a theory depend on the assumptions that underlie the theory, so they must empirically be checked.

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, and so on) solves a problem. This assertion is made implicitly or explicitly by almost all the papers. Descriptions of experiments also did not imply hypotheses if they served only to demonstrate an idea (for example, "The goal of the experiment is to make Genghis learn to walk forward" (799)). Worst-case complexity results did not count as predictions. As noted earlier, technically, they are predictions, but they predict nothing about average-case performance.

Only 25 papers contain anything that, by these criteria, could be called hypotheses or predictions. The others are 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 performing exploratory experiments to find out more about a functional relationship thought to underlie data. In general, if a paper goes beyond its central results or explains 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)

The following 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

Few papers discuss their results with any sense of surprise or discovery. Here are four 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)

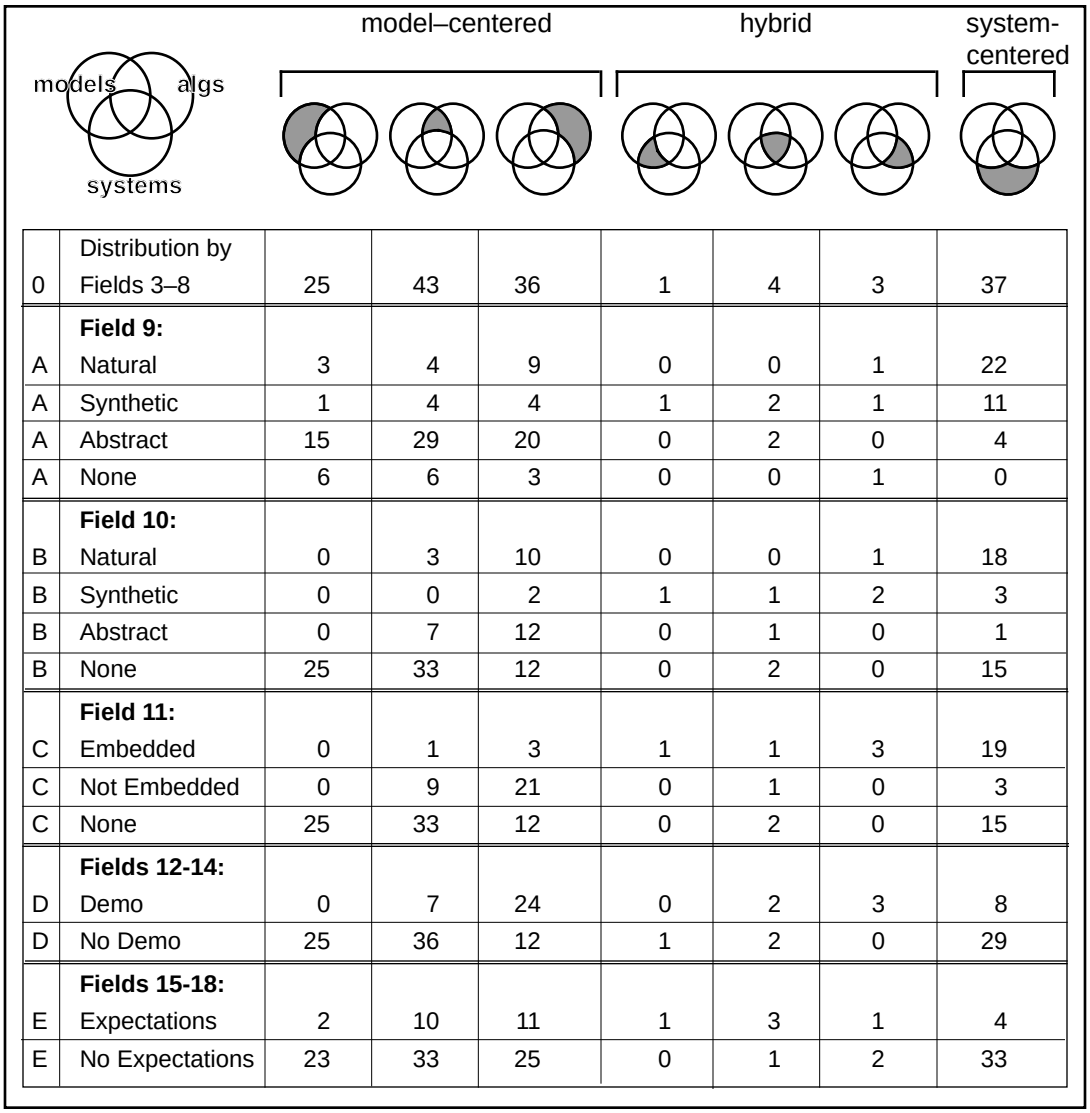
> An interesting result of our analysis is the discovery of a subtask that is at the core of generating explanations, and is also at the core of generating explanations in Reiter's default logic. (343)

> 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



0	Distribution by Fields 3–8	25	43	36	1	4	3	37
A	**Field 9:** Natural	3	4	9	0	0	1	22
A	Synthetic	1	4	4	1	2	1	11
A	Abstract	15	29	20	0	2	0	4
A	None	6	6	3	0	0	1	0
B	**Field 10:** Natural	0	3	10	0	0	1	18
B	Synthetic	0	0	2	1	1	2	3
B	Abstract	0	7	12	0	1	0	1
B	None	25	33	12	0	2	0	15
C	**Field 11:** Embedded	0	1	3	1	1	3	19
C	Not Embedded	0	9	21	0	1	0	3
C	None	25	33	12	0	2	0	15
D	**Fields 12-14:** Demo	0	7	24	0	2	3	8
D	No Demo	25	36	12	1	2	0	29
E	**Fields 15-18:** Expectations	2	10	11	1	3	1	4
E	No Expectations	23	33	25	0	1	2	33

Figure 3. Distributions of Papers by Classes and Fields.

include Hirschberg's test of an algorithm for assigning intonation to speech (previously discussed) and the following 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 appear in only four papers.

Statistical Analyses

The following analyses support the conclusions in section Hypothesis 1: Two Methodologies. Recall that papers were classified by fields 3–8 of table 1 into seven sets, shown in Figure 2. These sets are shown 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 labeled 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 give natural examples, 4 give synthetic examples, 29 give abstract examples, and 6 give no examples at all.

Now, although the 43 papers in MODELS ∩ ALGS are predominantly concerned with abstract examples, the 37 papers in SYSTEMS

	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.

	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.

(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 labeled A in figure 3, that is, 4 types of examples (including none) crossed with 7 sets of papers. Intuitively, the distribution seems unlikely to have occurred by chance; for example, I do not expect chance to distribute 15 of the 25 papers in MODELS into the abstract category and 22 of the 37 papers in SYSTEMS into the natural category. A chi-square test captures this intuition and tells whether the entire distribution (not only the anomalous examples that I pick out) could have arisen by chance. Given the contingency table in the rows labeled A in figure 3, a chi-square statistic (χ^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 (for example, MODELS or MODELS ∩ ALGS) a paper comes from; if the choice was independent of class, then the distribution would be expected by chance less than 1 time in 10,000.

The other way to see whether the distributions in figure 3 reflect different methodological tactics is to combine the original 7 sets of papers into 3: model centered, system centered, and hybrid. As shown at the top of figure 3, my scheme is papers in MODELS, ALGS, and MODELS ∩ ALGS are model centered (104 total); papers in SYSTEMS are system centered (37 total); and papers in MODELS ∩ ALGS ∩ SYSTEMS, MODELS ∩ SYSTEMS, and ALGS ∩ SYSTEMS are hybrid (8 total).

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

With this justification for the class of model-centered papers, the other classes naturally follow: 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 before, 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 labeled 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.

Similar analyses were run on fields 10 and 11, with the results reported in Hypothesis 1: Two Methodologies: Model-centered and system-centered papers focus on significantly different kinds of tasks and task environments. The data are shown in the rows labeled B and C, respectively, in figure 3. Note, however, that to analyze the embedded–non-embedded distinction, I constructed a contingency table that included only papers that describe a task (in fact, I left out the row labeled "C. None" in figure 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, I had to combine data from fields 15–18 into a single superfield called *expectations* (a yes in at least one of the fields counted as an expectation). The rows labeled E in figure 3 show the distribution of expectations. The contingency table for model-centered, system-centered, and hybrid papers was derived as previously described and shows that model-centered and hybrid papers are more likely than system-centered papers to discuss expectations ($\chi^2(2) = 10.5, p < .01$).

Second, I combined the data in fields 12–14 as shown in the rows labeled D in figure 3. These rows show the distribution of demonstrations over all papers. However, I also ran an

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

Figure 6. The Contingency Table for the Distribution of Demonstrations in Papers That Described Tasks over Three Classes of Papers.

Fields 4, 6 *or* 8:	Model-centered	Hybrid	System-centered
Analysis	82	6	16
No Analysis	22	2	21

Figure 7. The Contingency Table for the Distribution of Analyses over Three Classes of Papers.

analysis of the distribution of *demonstrations* over papers that describe tasks (field 10; see also rows B in figure 3). The contingency table in figure 6 shows that among the papers that describe a task, model-centered papers are more likely than system-centered papers to present a demonstration ($\chi^2(2) = 19.97, p < .001$).

Finally, to test whether model-centered or system-centered papers analyze their results to different extents, I had to slightly change the definitions of these classes. Recall that MODELS papers are those that present models (field 3) or prove theorems about the models (field 4). Let me 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 the 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 ($\chi^2(2) = 16.5, p < .0005$).

Problems with, and Concerns about, the Survey

It would be misleading to end this discussion without addressing some problems with my methodology, the way I conducted the survey. The major problem is that I have no reliability data. I 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 I 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 subjective; however, I needed some way to acknowledge the 13 papers that in an ill-defined way tried to analyze their systems (especially because 9 of them had no nonnegative entry in fields 12–18). I believe that other questions in table 1 can be answered more objectively but to find out requires a reliability study for which I solicit volunteers!

Bias because of preconceptions is another concern. Perhaps by identifying a paper as, say, a member of SYSTEMS, I became biased in how I filled in the other fields in table 1. For example, I 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 I expected SYSTEMS papers to demonstrate performance more often than ALGS papers. In fact, I expected exactly this result, but I 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 my results, but I doubt it is a major factor; at least, I have not discovered obvious examples of it.

I must also address the concern that the AAAI-90 papers 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 reviewing process. I 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 percent of the AAAI-90 papers provide 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. The fact that roughly 20 percent of the papers provide 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 belief, but it is not the simplest explanation of the wide variance in the quality of work presented at AAAI-90.

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). Thanks also to Adele Howe, Cynthia Loiselle, Scott Anderson, Dave Hart, and other members of the Experimental Knowledge Systems Laboratory. 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.

Notes

1. Mark Drummond pointed this possibility out.

2. Raj Reddy and other panelists at the recent Defense Advanced Research Projects Agency Workshop on Planning, Scheduling, and Control made this claim.

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

References

AAAI. 1990. Proceedings of the Eighth National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence.

Clancey, W. 1985. Heuristic Classification. *Artificial Intelligence* 27:289–350.

Cohen, P. R. 1991. Designing and Analyzing Strategies for PHOENIX from Models. In Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control, ed. K. Sycara, 9–21. San Mateo, Calif.: Morgan Kaufmann.

Cohen, P. R. 1990. Methodological Problems, a Model-Based Design and Analysis Methodology, and an Example. In Proceedings of the International Symposium on Methodologies for Intelligent Systems, 33–50. New York: North Holland.

Cohen, P. R. 1989. Evaluation and Case-Based Reasoning. In Proceedings of the Second Annual Workshop on Case-Based Reasoning, 168–172. San Mateo, Calif.: Morgan Kaufmann.

Cohen, P. R., and Howe, A. E. 1990. Benchmarks Are not Enough; Evaluation Metrics Depend on the Hypothesis. Presented at the Workshop on Benchmarks and Metrics, Mountain View, Calif.

Cohen, P. R., and Howe, A. E. 1988a. How Evaluation Guides AI Research. *AI Magazine* 9(4): 35–43.

Cohen, P. R., and Howe, A. E. 1988b. Toward AI Research Methodology: Three Case Studies in Evaluation. *IEEE Transactions on Systems, Man, and Cybernetics* 19(3): 634–646.

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

De Mey, M. 1982. *The Cognitive Paradigm*. Boston: Reidel.

Heckerman, D. 1986. Probabilistic Interpretations for MYCIN's Certainty Factors. In *Uncertainty in Artificial Intelligence*, eds. L. Kanal and J. Lemmer, 167–196. Amsterdam: North-Holland.

Howe, A. E.; Hart, D. M.; and Cohen, P. R. 1990. Addressing Real-Time Constraints in the Design of Autonomous Agents. *The Journal of Real-Time Systems* 1:81–97.

Korf, R. 1987. Planning as Search: A Quantitative Approach. *Artificial Intelligence* 33:65–88.

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, 109–114. San Mateo, Calif.: Morgan Kaufmann.

Langton, C. 1989. *Artificial Life*. Santa Fe Institute Studies in the Sciences of Complexity. Reading, Mass.: Addison-Wesley.

Lenat, D. B., and Feigenbaum, E. A. 1987. On the Thresholds of Knowledge. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1173–1182. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Levesque, H. J., and Brachman, R. J. 1985. A Fundamental Trade-Off in Knowledge Representation and Reasoning (revised version). In *Readings in Knowledge Representation*, eds. R. J. Brachman and H. J. Levesque, 41–70. San Mateo, Calif.: Morgan Kaufmann.

Lifschitz, V. 1987. On the Semantics of STRIPS. In *Reasoning about Actions and Plans*, eds. M. Georgeff and A. Lansky, 1–9. San Mateo, Calif.: Morgan Kaufmann.

McDermott, D. 1981. Artificial Intelligence Meets Natural Stupidity. In *Mind Design*, ed. J. Haugeland, 143–160. Montgomery, Vt.: Bradford.

Mitchell, T. M. 1981. Generalization as Search. In *Readings in Artificial Intelligence*, eds. B. L. Webber and N. J. Nilsson, 517–542. San Mateo, Calif.: Morgan Kaufmann.

Paul Cohen is an associate professor in the Department of Computer and Information Science at the University of Massachusetts at Amherst, where he directs the Experimental Knowledge Systems Laboratory. Cohen and his students are currently working on a book that documents research in the MAD methodology, using the PHOENIX fire-fighting simulation as a testbed. Cohen is also an editor of the *Handbook of Artificial Intelligence*, Volumes III and IV.