A 4-Space Model of Scientific Discovery

Christian Schunn & David Klahr
Department of Psychology, Carnegie Mellon University

One fruitful characterization of scientific discovery is to view it in terms of search in two problem spaces: a space of hypotheses and a space of experiments (Klahr & Dunbar, 1988; Simon & Lea, 1974). This characterization can be used to classify discovery models into three groups. First, there are those that address the processes of hypothesis generation and evaluation (e.g., the BACON models & variants (Langley, Simon, Bradshaw, & Zytkow, 1987), IDS (Nordhausen & Langley, 1993), PHINEAS (Falkenhainer, 1990), COPER (Kokar, 1986), MECHEM (Valdés-Pérez, in press), HYPGENE (Karp, 1990, 1993), AbE (O'Rorke, Morris, & Schulenburg, 1990), OCCAM (Pizzani, 1990), and ECHO (Thagard, 1988)). Second, there are those that address the process of experiment generation and evaluation (e.g., DEED (Rajamoney, 1993), and DIDO (Scott & Markovitch, 1993)). Third, there are those that address both processes (e.g., KEKADA (Kulkarni & Simon, 1988), STERN (Cheng, 1990), HDD (Reimann, 1990), IE (Shrager, 1985), SDDS (Klahr & Dunbar, 1988), and LIVE (Shen, 1993)).

Based on our analysis of subject's performance in a complex computer microworld, we have extended the two-space framework to a four-space framework. In the new framework, what was previous conceived as the hypothesis space has now been divided into a *data representation space* and a *hypothesis space*. In the data representation space, representations or abstractions of the data are chosen from the set of possible features. In the hypothesis space, hypotheses about causal relations in the data are drawn using the set of features in the current representation. Similarly, the old experiment space is now divided into an *experimental paradigm space* and an *experiment space*. In the experimental paradigm space, a class of experiments (i.e., a paradigm) is chosen which identifies the factors to vary, and the components which are held constant. In the experiment space, the parameters settings within the selected paradigm are chosen.

We made these changes as we began to scrutinize the human performance data from several discovery microworlds in preparation for the computational implementation of the two-space model. It became clear that, during the course of their investigations of the domain, subjects often acquired new data representations, and developed new kinds of experiments. Furthermore, representation and paradigm selection appear to require different mechanisms from those necessary for hypothesis and experiment selection.

Our goal is to produce a unified model of processing in all four problem spaces. This unified model will consist of separate components corresponding to processing in each of the four problem spaces. However, as indicated in Figure 1, processing within each space is dependent upon the current search state in the other spaces. For example, experiment space search depends upon the available experimental paradigms as well as the current hypothesis. Given these strong interdependencies, there is great advantage to implementing each of the components in a unified model.

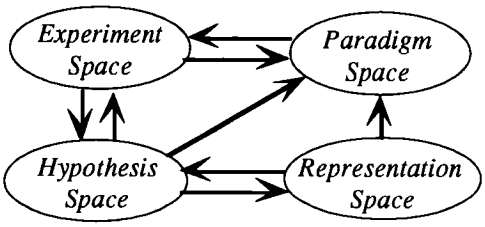

Figure 1. The informational interdependencies among the four search spaces. The arrows indicate the direction of information flow.

Before presenting the details of the model, we will provide a brief description of the task domain and data that lead to the new model.

The task. The task that contributed the primary data for our model design is a complex computer microworld called MilkTruck (Schunn & Klahr, 1992, 1993, 1995). In the MilkTruck domain, subjects conducted experiments to discover the action of a complex mystery function. In the microworld, a "milk truck" executed a sequence of actions associated with a dairy delivery route. At any of 6 different locations along its route, it could beep its horn, deliver milk or eggs, or receive money or empties. A program consisted of a sequence of up to eight action-location pairs. After the route had been entered, the subject pressed 'RUN' and the milk truck executed its route on the screen. The milk truck went to each location on the programmed route in the order that it was programmed, and the subjects were shown by way of animated icons what transpired at the location.

In this task, subjects were given a great deal of external memory support. As subjects entered their programs, the steps were displayed on the screen in the program listing. Also, as the route was completed, a trace listing displayed in program format what transpired during the run (see figure 2). The subjects were also given access to all previous programs and traces.

When the mystery command, δ (delta), was not used, the trace listing was identical to the program listing. However, the δ command could change the order of delivery, and the resultant route execution and its associated

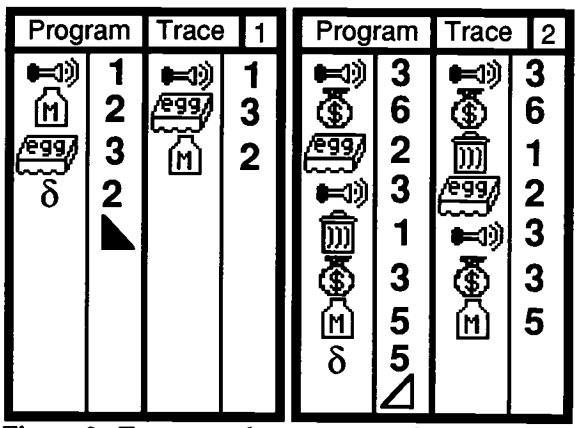

Figure 2. Two example programs and outcomes.

For the last N steps in the program, δ reorders the execution sequence of the program by...	
◿ (increasing)	...increasing house number order.
◺ (decreasing)	...decreasing house number order.

Table 1. The function of the δ arguments.

trace would then be discrepant from the program listing. The effect of the δ command was to reorder the execution of part of the program according to the values of its two arguments, a number (1-6), and a triangle (white or black). Table 1 describes the effects of the δ command.

In one version of the MilkTruck task, there was a third parameter, a Greek letter, (α or β). When β was chosen, the δ worked as above. However, when α was chosen, the deliveries were reordered by delivery item order rather than by delivery location. In this version, programs could have up to 14 steps.

The subjects were Carnegie Mellon University undergraduates, coming from a mix of science, engineering, and arts backgrounds. Subjects typically took part in a single, one hour session. Following an introduction to the basics of the MilkTruck domain, the syntax of the δ command was described, and the goal of discovering the effect of the δ command was presented to the subjects. In the discovery phase, subjects designed, conducted, and analyzed experiments with the goal of discovering the role of the δ command and its arguments.[1] The subjects worked at the discovery task until they felt that they had solved it, or they wished to give up.

The number of potential experiments in MilkTruck domain is 8.9×10^{12} in the 2 parameter version and 1.2×10^{22} in the 3 parameter version. Thus, subjects were forced to be extremely selective in deciding which experiments to run. In the two parameter MilkTruck task, subjects ran approximately 20 experiments across 30 minutes. In the three parameter MilkTruck task, subjects ran approximately 50 experiments across 80 minutes.

The data. Data was collected from over 100 subjects across various conditions involving the two versions of the MilkTruck task. Both key stroke and verbal protocols were collected from all of the subjects. Here, we present a very brief description of the kinds of data from these experiments that contributed to the creation of the 4 space model (see Schunn & Klahr, 1991, 1992, 1995 for further detail). In particular, we will focus on the evidence which suggested the addition of the experimental paradigm and data representation spaces.

The primary evidence for activity in the data representation space involved changes in subjects' descriptions of experimental outcomes. Early in the sessions, subjects typically described experimental outcomes in terms of series of movements of single steps (e.g., the second step moved to the fourth position and the third step moved to the second position). Later in the sessions, subjects began to give descriptions for, and hypothesize about, the same kinds of experimental outcomes in terms of movements of segments of the program (e.g., the last three steps were reorganized; the whole program was reversed). While one might argue that these changes were merely redescriptions or reorganizations of the same features, some subjects also added completely novel features to their descriptions (e.g., the number of times the milk truck changed directions during the delivery route; the number of times the milk truck driver jumped up and down at the end of the route). This kind of evidence led us to hypothesize that subjects changed the way in which the basic data was used by adding and deleting features to their data representation.

The primary evidence for activity in the experimental paradigm space derived from subject's stated plans for their experiment selections and changes in these plans over the course of the problem-solving session. Initially, subjects had very few kinds of experiments from which to select. Their primary decisions involved which items to use and which δ parameter combination to try next. Later in the session, subjects began to develop more complex kinds of experiments. For example, a subject might design a five step program using the same delivery item in each step, with houses in decreasing number order. Subjects also developed multi-program paradigms. For example, a subject might decide to conduct a sequence of five programs with the same base program, varying only the δ number parameter. Subjects learned to generate these complex, very deliberately chosen experiments quite rapidly, indicating that their were choosing experiments from a newly compiled database of experiment types.

The model. The entire model is being implemented in ACT-R (Anderson, 1993), a production system architecture that embodies a theory of human cognition. One advantage of using ACT-R is that it has learning capabilities. Another is that, because it is intended as a theory of human cognition, ACT-R's treatment of low-level processes (e.g., memory) is already empirically constrained. This is desirable in psychological models of discovery, since data about high-level processes like discovery phenomena are not sufficiently detailed for making implementation decisions about low-level processes. A brief description of the processes propose within each of the spaces, and how they will be implemented in ACT-R is presented below (see Table 2 for a list of the processes in the model).

The experimental paradigm space. On occasion, making an important discovery involves finding a new

Space	Process
Hypothesis	representational mapping pop-out piecemeal induction
Data Representation	notice-invariants analogy divergent search
Experiment	complexity management risk regulation theory orientation
Experimental Paradigm	analogy error analysis hypothesis testing rep/ hyp change

Table 2. The component processes within each of the search spaces of the 4 space model.

[1] In this context, a program is an "experiment" and a statement abot how the parameters work is a "hypothesis".

method for gathering data—a new experimental paradigm.[2] It is unlikely that his new method for gathering data is some new domain-general induction method (e.g., Mill's inductive cannons); instead it is likely to be a method unique to that field of inquiry (e.g., changing the temperature in a particular order, instructing subjects in a particular way). These developments are typically not new instruments being developed (although they can be); rather they are typically new methods for using the same instruments. The issue at hand is how such new methods are created. Our model includes several domain-general heuristics for the creation of such new methods.

Paradigms are primarily created in the service of testing a hypothesis. The hypothesis embodies a set of assumptions about what features of the experiment are of interest. An experimental paradigm is created that emphasizes the features of interest. For example, to test the hypothesis that wing-span matters to airplane lift capacity, an experimenter might create a paradigm in which wing-span is easily manipulated. The corollary of this paradigm creation process is that paradigms are also created to de-emphasize features which are hypothesized not to matter (either by holding those features constant, or by removing them entirely from the experiment).

Paradigms can also be created through analogy to other paradigms. For example, subjects in the MilkTruck task developed the paradigm of holding delivery item constant from the paradigm of holding house number constant. These analogous, example paradigms can be ones acquired through observation, or ones generated by oneself in other situations. Paradigms may also be created by analyzing the cause of failed experiments. For example, if an experiment produces an ambiguous outcome, a new paradigm can be created to disambiguate the outcome. Furthermore, these new paradigms may be created through an error analysis of thought experiments rather than actual experiments.

In the ACT-R implementation of the model, the experimental paradigm library is represented with productions. Different features of the environment (e.g., the current hypothesis, the existence of an alternative hypothesis, the current data representation) either trigger decisions about which experiment features are held constant, or set goals to make decisions about certain features of the experiment. These productions are created using domain-general heuristics, and by analogy to other such productions.

The experiment space Many costs and risks are associated with conducting experiments (e.g., mental effort, time spent, money spent, and potential loss of face for a failed experiments), and one practical goal of experimentation is to minimize these costs and risks. Experimentation also has theoretical goals related to the acquisition of information about the world. For example, it is desirable to design experiments relevant to the question at hand, with easily-interpreted and unambiguous results. How are these multiple and often conflicting goals achieved in particular experiments?

In our model, the theoretical goals of experiment selection are achieved using two main heuristics: the *examination heuristic* and the *discrimination heuristic*. The examination heuristic selects experiments which directly demonstrate the hypothesized effect. For example, a hypothesis about the behavior of acids in the presence of water leads to the selection of an experiment involving water. This tendency produces what has been called the +H test bias (Klayman & Ha, 1987) in rule discovery tasks: rules of the form "X's are a member of the concept" will lead to the selection of X's, rather than things that are not X's, to test the rule. The discrimination heuristic selects experiments which can discriminate among competing hypotheses under consideration. This heuristic is used only when multiple hypotheses are being considered. Therefore, there is no bias to select highly discriminating experiments (experiments which discriminate among many potential hypotheses) in the absence of multiple, specific, active hypotheses. However, risk regulation (see below) does take into account expected information content.

The practical goals of experiment selection are met through processes of *complexity management* and *risk regulation*. *Complexity management* involves regulating experiment design complexity and experiment interpretation complexity, where complexity is defined relative to the current state of understanding and experimental expertise. For example, longer experiments are more difficult to generate when few operators for generating long experiments exist, and the longer experiments are more difficult to interpret when the knowledge of relevant dimensions is small.

Risk regulation involves choosing experiments based on their perceived probability of producing an informative outcome. In many cases, this involves choosing between experiments which have a low probability of being successful[3], yet would be very informative if they are successful, and experiments which have a high probability of being successful even though they contain little potential information. For example, conducting experiments which vary few features from the previous experiment are likely to behave exactly as predicted, whereas experiments in which many features have been varied have the potential of producing very novel results yet carry the risk of producing uninterpretable results.

Complexity management and risk regulation are often in opposition. For example, more complex experiments are more likely to be informative, but are also much more difficult to generate and interpret. These two factors are combined to produce an expected utility, which determines the final experiment choice. The balance between complexity management and risk regulation varies with expertise. For example, with experience, longer programs become more easily generated and interpreted.

[2]The most popular use of the term "paradigm", typically associated with Kuhn (1970), refers to a much larger entity than we are considering. In fact, Kuhn used the word "paradigm" in two senses (which he acknowledges in the postscript of the second edition): the large scale paradigm of a whole field, and the smaller scale experimental paradigms that are used in particular experiments (e.g., the paired-associates paradigm, or Sperling's iconic memory paradigm). We will use the term to refer to experimental paradigms of the second, smaller, kind.

[3] A successful experiment is one that can be meaningfully predicted or postdicted.

In the ACT-R implementation of the model, the discrimination and examination principles are implemented primarily with domain-general productions, although they are supported by hypothesis-specific mental simulation productions. The complexity management and risk regulation processes make use of the rational conflict resolution in ACT-R. That is, different experiment selections are represented with different production instantiations. Each instantiation has an expected probability of success (i.e., leading to a successful experiment) and an expected cost. ACT-R's conflict resolution algorithm then selects the production with the best tradeoff between expected cost and success. These expected values are computed using simple a priori assumptions about how expected cost and probability of success vary with the choice of experiment features. For example, in the MilkTruck task, expected cost is assumed to be a linear function of program length. The constants involved in these a priori functions are estimated continuously during the task. Thus, through feedback from the actual costs and success rates, the model learns how much weight to place on each of the various factors.

The data representation space. How does one choose or change a data representation? Finding the general solution to these questions is a difficult task because there is no known universal language for describing data representations, nor is there a known general generator of representations. As a partial solution to these questions, we present three heuristics used for selecting representations from a previously existing repertoire (i.e., no entirely new representations are created).

In our model, data representation change occurs through the following mechanisms: *Notice Invariants*, *Analogy*, and *Brute-force search*. *Notice Invariants* works as follows. Experience with experimental outcomes within a domain leads to the noticing of certain regularities. New representations are chosen which emphasize these regularities. For example, a scientist might notice that biplanes have the greatest degree of stability when the two sets of wings are both large or both small. As a result, the scientist may adopt a representation which includes wing-set size-differences. This behavior is exemplified in the MilkTruck task as subjects begin to notice that the first part of the program rarely changes. They then change their data representations to include changing and unchanging segments of the program.

Analogy produces representations by analogy to previously understood phenomena. For example, such analogies might include: computers are like programmable calculators, and atoms are like the solar system. The features used in the analogical source are applied to the analogical target. This process is similar to a categorization process. One kind of situation that triggers this process is the occurrence of salient, expectation-violating events, which force the recategorization of objects and events.

Brute-force search is a process of searching haphazardly through the set of possible representations of objects in the environment (i.e., by considering each object, and all the features and feature clusters of each object). This is the method by which subjects in the MilkTruck domain tended to add features to their data representations. The order of search may be constrained by the salience or availability of the possible representations. The process of brute-force search typically occurs when the individual believes that the current representation may not include the causally-predictive features.

In the ACT-R implementation of the model, many of these processes involve a great deal of domain-specific knowledge. The Notice-Invariants process involves domain-specific productions for noticing various kinds of features. The Analogy and Brute-force search processes involve domain-specific declarative knowledge of analogical sources and properties of objects. However, there are also a few domain-general processes that focus attention on invariant features, that search for analogies, and that decide to begin a Brute-force search.

The hypothesis space. In our model, the fundamental character of search in the hypothesis space is the piece by piece construction of hypotheses. This process is called *piecemeal induction*. In the first stage of piecemeal induction, a hypothesis is generated (either from memory or from data). Then, a scoping processes determines the generality or scope of the hypothesis. For example, a biologist might hypothesize that an enzyme functions in a certain manner when the ambient temperature is between 40F and 50F (in contrast to concluding that the enzyme functions in that manner no matter what the temperature). The dimensions used to form the scope are chosen from the current data representation. On each dimension, the most general scope value is preferred in the absence of counter-evidence.

With one or more particular hypotheses as input, abstraction processes generate more general hypotheses. For example, in the MilkTruck domain, subjects often abstract the hypothesis that the last N steps of the program are reordered from the particular hypotheses that there is no change with N=1, and the last two steps are changes with N=2. The number of particular cases that are sufficient to warrant a generalization is dependent upon expectations about the variability in the domain of study (which can be modified with experience).

There are many candidate mechanisms for the generation of hypotheses. In the domains that we have studied, two main processes have been found: *representational mapping*, and *pop-out*. *Representational mapping* is mapping of objects onto actions or parts of actions, where both the object and the actions (and action parts) are already in the current representation. Representational mapping is similar to a memory search. The representations in memory are searched for correspondences. The more complex the mapping (i.e., greater number of predicates), or the less salient the to-be-mapped feature, the lower the probability that the mapping will occur.

Representational mapping uses two heuristics: unique-function, and same-type. The unique-function heuristic favors mapping objects with no other known function onto actions or components of actions with no other known cause. The same-type heuristic favors mapping objects onto things of the same dimensionality. For example, binary object factors (e.g., feature presence) tend to be mapped onto inherently binary output factors (e.g., effect presence). No actual experimental outcomes are necessary

for representational mapping, since representational mapping can work with abstract schemata as well as particular objects. Therefore, this mechanism is typically used for generating initial hypotheses in the absence of evidence.

Pop-out occurs through automatic, categorization processes. When certain evidence presents itself, certain relationships are uniformly entertained. For example, exact similarity (whether coincidental or not) is automatically noticed. This automatic process is dependent upon representational factors. For example, if a feature is not encoded, no similarity involving that feature can be noticed.

In the ACT-R model, the implementation of the hypothesis space search processes is quite complex, and only very briefly summarized below. A hypothesis is represented both by a declarative representation of the current status of the hypothesis (e.g., tested and confirmed, tested and refuted, or untested), and by a set of productions that recognize which experimental outcomes support or refute the hypothesis. The pop-out processes involve low-level perceptual processes, also represented with productions. The heuristics used in the representational mapping process are enforced by the ACT-R architecture itself—the production creation mechanism has similar constraints. Finally, the abstraction and scoping processes involve several domain-general heuristics implemented in productions.

Comparison to previous work. The details of our model are similar in many respects to other discovery models. The search space with the greatest degree of similarity is the hypothesis space. In particular, our piecemeal induction processes are very similar to the quantitative and qualitative rule induction processes of the BACON models (Langley, et al., 1987). FAHRENHEIT (Zytkow, 1987) is the intellectual precursor of our scoping processes. The units checking processes in COPAR (Kokar, 1986) and ABACUS (Falkenhainer & Michalski, 1986) that constrain allowable hypothesis are similar in effect to the same-type heuristic. The unique-function heuristic is used in several inductive systems, including LIVE (Shen, 1993) and PUPS (Anderson & Thompson, 1989). IE (Shrager, 1985) uses a component of this heuristic, the unoverlapping heuristic, to insure that objects are hypothesized to have a most one function.

The pop-out mechanism we use is a very generic computational principle. Many production systems models have domain-specific productions which immediately recognize and hypothesize about certain kinds of relations and correspondences. For example, KEKADA (Kulkarni & Simon, 1988) immediately recognizes mixed or additive effects given certain kinds of data. In another domain, STERN (Cheng, 1990) immediately recognizes power functions in quantitative data. The pop-out of identity and item-order relations that our model uses has precursors in the letter series completion models of Simon & Kotovsky (1963) and Klahr & Wallace (1972).

Turning to experiment space processes, there are no models of discovery that explicitly address the issue of complexity management. In contrast, several discovery systems have methods for ordering the experiment space search such that experiments likely to produce useful information are considered first. For example, AM (Lenat & Brown, 1984) and EURISKO (Lenat, 1983) focus attention on concepts that produce novel results. In a similar fashion, DIDO (Scott & Markovitch, 1993) uses a curiosity heuristic which favors experiments testing the maximally uncertain part of the hypothesis. However, DIDO regulates whether experimental outcomes are considered further or ignored rather than regulating which experiments are conducted.

The examination principle is implicit in many models (e.g., LIVE (Shen, 1993), AM (Lenat & Brown, 1984), EURISKO, DIDO (Scott & Markovitch, 1993), and DEED (Rajamoney, 1993)), but explicit in IE (Shrager, 1985) only. The discrimination principle is also taken from Shrager's IE model. However, there are similar principles in several other systems, including DEED (Rajamoney, 1993), and ABD-Soar (Johnson, Krems, & Amra, 1994).

Very few discovery systems create new experimental paradigms, and fewer still have considered this search space explicitly. STERN (Cheng, 1990) is one of the few such models. It uses a different method of paradigm creation that the methods found in our model; STERN constructs new paradigms by combining existing experimental paradigms such that the output (dependent) values of one paradigm are used as input (independent) values of another. The most important difference between the paradigm construction in STERN and our model is that STERN creates new paradigms simply to try something new, whereas our model creates new paradigms because some feature of the new paradigm is desired.

Data representation change also has rarely been modeled. However, Kaplan's SWITCH (1989) presents a few heuristics for representation change, and they are different from the three heuristics explicitly postulated here (e.g., change grain size on failure, and pursue hot ideas). Furthermore, there are many programs which are capable of proposing new intrinsic properties, a kind of invariance. For example, BACON.4 (Langley et al., 1987) discovers the intrinsic property gravitational mass from the properties of force and distance by searching for constant relations among factors. There are also several kinds of conceptual hierarchy discovery programs that discovery new categories (i.e., new representations) by searching for feature invariance (e.g., Fisher's COBWEB (1987)). Two models produce new representations via analogy, SWITCH (Kaplan, 1989) and View Application (Shrager, 1987). However, they do not have a method for choosing analogs (i.e., the analog must be given to the model). Finally, turning to the Brute-force search process, there are no previous models which consider random data representations. There are, however, several models which focus attention on random portions of the existing data representation (e.g., LIVE (Shen, 1993) and EURISKO (Lenat, 1983)).

Conclusion. We have presented a general framework for understanding scientific discovery: the 4-space framework of experimental paradigm, experiment, data representation, and hypothesis. This framework is a significant extension to the experiment and hypothesis space focus of many previous models of discovery, and we expect it to be applicable to many discovery domains.

In this framework, we also have provided an outline for how these four spaces interact. We suggest that since

processing in these four spaces can be highly interdependent, there are theoretical and computational advantages associated with considering all four spaces. In particular, previous models of discovery may have been trying to solve the difficult problems of data representation and experimental paradigm search in the process of dealing with hypothesis and experiment space issues, and may have been confounding separate issues in the process. By considering these issues as conceptually distinct factors, and by studying their interrelations, we may gain further insight into the automation of scientific discovery.

In addition to providing the general 4 space framework, we have also presented descriptions of the component processes. These component processes are implemented in a current, and detailed cognitive architecture, ACT-R. The advantages of using a cognitive architecture extend beyond greater fidelity to human performance: the resulting models of discovery are pushed towards using more efficient discovery heuristics because the cognitive architectures typically assume limited computational capacities (in both time and space). For example, in our model the complexity management and risk regulation processes are implemented as simple derivatives of the conflict resolution algorithm.

The goal of our future computational work will be to pursue the complete implementation of our model, and assess the tractability of our theoretical model, as well as its generalizability to other domains.

References

Anderson, J. R. (1993). *The rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Thompson, R. (1989). Use of analogy in a production system architecture. In S. Vosniadou, & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 267-297). Cambridge, MA: Cambridge University Press.

Cheng, P. C.-H. (1990). *Modeling scientific discovery.* Unpublished doctoral dissertation, The Open University, Milton Keynes.

Falkenhainer, B. C. (1990). A unified approach to explanation and theory formation. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation.* San Mateo, CA: Morgan Kaufmann.

Falkenhainer, B. C., & Michalski, R. S. (1986). Integrating quantitative and qualitative discovery: the ABACUS system. *Machine Learning*, **1**(4), 367-401.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, **2**, 193-211.

Johnson, T. R., Krems, J., & Amra, N. K. (1994). A computational model of human abductive skill and its acquisition. In *Proceedings of the 16th Annual Conference of the Cognitive Science Society.* LEA.

Kaplan, C. A. (1989). *SWITCH: A simulation of representational change in the Mutilated Checkerboard problem* (Tech. Rep. No. 477). Pittsburgh: Carnegie Mellon University, Department of Psychology.

Karp, P. (1990). Hypothesis formation as design. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation.* San Mateo, CA: Morgan Kaufmann.

Karp, P. (1993). Design methods for scientific hypothesis formation and their applications to molecular biology. *Machine Learning*, **12**, 89-116.

Klahr, D. & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science*, **12**, 1-48.

Klahr, D. & Wallace, J. G. (1970). The development of serial completion strategies: An information processing analysis. *British Journal of Psychology*, **61**, 243-257.

Klayman, J. & Ha, Y.-W. (1987). Confirmation, disconfirmation, and information in hypothesis testing. *Psychological Review*, **94**(2), 211-228.

Kokar, M. M. (1986). Determining arguments of invariant functional descriptions. *Machine Learning*, **1**(4), 403-422.

Kuhn, T. S. (1970). *The structure of scientific revolutions*, 2nd edition. Chicago: University of Chicago Press.

Kulkarni, D. & Simon, H.A. (1988). The process of Scientific Discovery: The strategy of Experimentation. *Cognitive Science*, 12, 139-176.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery: Computational explorations in the creative process.* Cambridge, MA: MIT Press.

Lenat, D. B., & Brown, J. S. (1984). Why AM and EURISKO appear to work. *Artificial Intelligence*, **23**, 269-94.

Lenat, D. B. (1983). EURISKO: A program that learns new heuristics and domain concepts. *Artificial Intelligence*, **21**, 68-98.

Nordhausen, B., & Langley, P. (1993). An integrated framework for empirical discovery. *Machine Learning*, **12**, 17-47.

O'Rorke, P., Morris, S., & Schulenburg, D. (1990). Theory formation by abduction: A case study based on the chemical revolution. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation.* San Mateo, CA: Morgan Kaufmann.

Pizzani, M. J. (1990). *Creating a memory of causal relationships.* Hillsdale, NJ: Erlbaum.

Rajamoney, S. A. (1993). The design of discrimination experiments. *Machine Learning*, **12**, 185-203.

Reimann, P. (1990). Problem solving models of scientific discovery learning processes. Frankfurt am Main: Peter Lang.

Schunn, C. D., & Klahr, D. (1992). Complexity Management in a Discovery Task. In *Proceedings of the 14th Annual Conference of the Cognitive Science Society.* Cambridge, MA: MIT Press.

Schunn, C. D., & Klahr, D. (1993). Self- Vs. Other-Generated Hypotheses in Scientific Discovery. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Schunn, C. D., & Klahr, D. (1995). Complexity management and risk regulation in scientific discovery. Unpublished manuscript.

Scott, P. D., & Markovitch, S. (1993). Experience selection and problem choice in an exploratory learning system. *Machine Learning*, **12**, 49-67.

Shen, W-M. (1993). Discovery as autonomous learning from the environment. *Machine Learning*, **12**, 143-165.

Shrager, J. C. (1985). *Instrucionless learning: Discovery of the mental model of a complex device.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh.

Shrager, J. C. (1987). Theory change via View Application in instructionless learning. *Machine Learning*, **2**, 247-276.

Simon, H. A. & Kotovsky, K. (1963). Human acquisition of concepts for sequential patterns. *Psychological Review*, **70**, 534-46.

Simon, H. A., & Lea, G. (1974). Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), *Knowledge and cognition.* Hillsdale, NJ: Erlbaum.

Thagard, P. (1988). *Computational philosophy of science.* Cambridge, MA: MIT Press.

Valdés-Pérez, R. E. (in press). Conjecturing hidden entities via simplicity and conservation laws: Machine discovery in chemistry. *Artificial Intelligence.*

Zytkow, J. M. (1987). Combining many searches in the FAHRENHEIT discovery system. In Proceedings of the Fourth International Workshop on Machine Learning. Irvine CA, 281-87.