Prolegomena to a Task-Method-Knowledge Theory of Cognition

J. William Murdock (murdock@cc.gatech.edu)
College of Computing
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, Georgia 30332-0280, USA

Abstract

How can we integrate interrelated theories of individual elements of cognition? Computational models of reasoning processes encode an *understanding* of reasoning. Consequently, a computational modeling language may be ideally suited to the presentation of theories of cognition. By representing theories of a variety of phenomena in a single modeling language, we can potentially explore how these theories might interact. The Task-Method-Knowledge (TMK) modeling language evolves from artificial intelligence research on the subject of multistrategy reasoning. TMK models provide a *compositional* account of reasoning processes; they describe not only what the elements of a process are, but also how the functional properties of these elements combine to form the functional properties of the process as a whole. This paper explores the composition of theories of cognition within the TMK framework, drawing on some existing theories within cognitive science as examples.

Introduction

The goal of cognitive science is the development of models of the functionality of the human mind. The vast majority of research in this field has focused on the development of specific models of particular phenomena such as pattern recognition, short-duration remembering, dual-task performance, etc. The underlying assumption in such research is that these individual pieces of a general model of human intelligence could potentially be combined to form a complete model. This approach raises the issue of how such an integration might be accomplished.

Two very successful theories of a general framework for models of cognition are SOAR (Laird et al., 1987) and ACT (Anderson, 1983; Anderson, 1993). Both of these theories model cognitive capabilities as production rules. Integration of reasoning strategies within these frameworks is done by combining sets of production rules. This provides a parsimonious environment for combining models of various aspects of cognition. Such systems generally provide a powerful mechanism for modeling precisely *what* a mind is doing. However, such systems do not provide higher level abstractions which combine individual productions into more complex units of functionality. Consequently, I claim, production rule systems are very limited in the extent to which they represent *how* and *why* a mind does what it does. Thus a model in these frameworks may not convey a full understanding of a phenomenon.

What does it mean to understand a complex system, such as the mind? One answer to this question is suggested by a line of research which examines computational representations of physical devices (Goel, 1989; Bhatta, 1995; Goel et al., 1996; Goel et al., 1997; Griffith, 1997). In this work, it is shown that a wide variety of reasoning tasks relating to physical devices can be supported by models which are causal (i.e., that show the mechanisms by which effects occur), compositional (i.e., that show how the effects of the separate elements of a device are combined), and functional (i.e., that take an intentional stance toward describing why elements are arranged as they are). The basic idea behind these models is that the relationship between the physical construction of a device and the intended effect of that device is described by a flow of causal interactions describing the device's behavior. Because these traits have been shown to produce models which enable a very broad range of reasoning tasks, I argue that they form an example of true comprehension. In other words, the important contribution that this body of work makes to the argument I am making here is:

> An accurate causal, compositional, functional model of a complex system *inherently constitutes* a deep understanding of such a system.

The Task-Method-Knowledge (TMK) modeling language (Stroulia, 1994; Goel et al., 1996; Griffith, 1997; Murdock and Goel, 1998) provides a causal, compositional, functional framework for describing cognitive capabilities. TMK models have been used to support processes such as explanation and adaptation in a variety of AI systems. For example, (Murdock and Goel, 1998) describes a recently developed agent architecture for implementing, executing, and adapting TMK models. TMK models are very much an extension of the physical device modeling framework from which they were derived. The division of reasoning into tasks (i.e., functional elements) and methods (i.e., behavioral elements) very much duplicates the functional and causal features of the physical device models, and the explicit modeling of knowledge states duplicates the compositional aspects of these models.

I claim that the TMK language provides a useful framework for integrating models of cognitive processes because a comprehensive model presented within this framework can convey a deep understanding of cognition. This paper investigates this claim.

Technical Details

The three basic elements of TMK models are tasks, methods, and knowledge. Tasks are units of functionality; they represent *what* is done. Methods are units of behavior; they represent *how* something is done. Tasks and methods are intimately interconnected; tasks are linked to a set of methods which accomplish those tasks, and methods are in turn linked to a set of lower-level tasks which are necessary to accomplish that method. Consider for example, the mathematical task of adding a set of three digit numbers. This task might be accomplished by a standard long addition algorithm, which woulded call a method in the TMK language. This method is defined by the subtasks which accomplish it, i.e., adding of columns and carrying of remainders, ordered in an iterative loop. The task of adding a column might, in turn, be accomplished by one of several methods, e.g., direct memory retrieval, counting on one's fingers, etc. The knowledge portion of the TMK models provides the language in which the requirements of the tasks and the capabilities of the methods are defined; for the addition task, the knowledge portion of the TMK model might describe concepts such as numbers, digits, columns, etc. as well as relationships such as sums of digits, adjacency of columns, etc.

Consider a more elaborate example: qualitative, conceptual design of physical devices. The KRITIK series of systems (Goel, 1989; Goel et al., 1997; Goel et al., 1996) instantiates a theory of this reasoning process. Figure 1 shows a few of the highest levels of a TMK model of design inspired by KRITIK. This figure describes a design task with two top level methods: case-based reasoning and generate and test. The case-based reasoning method involves retrieval, adaptation, verification, and storage. The generate and test method involves generation (e.g., by following simple design heuristics) followed by verification. The task of design verification, which is common to the two top-level methods, is further elaborated by two lower-level methods: qualitative simulation and physical instantiation. The qualitative simulation method involves tracing through the design to make certain that the device should accomplish the specified function. The physical instantiation method involves actually building the device and seeing if it operates as specified (this method illustrates a crucial feature of TMK models; they seamlessly integrate reasoning and action by allowing tasks to be accomplished by both reasoning strategies and action strategies).

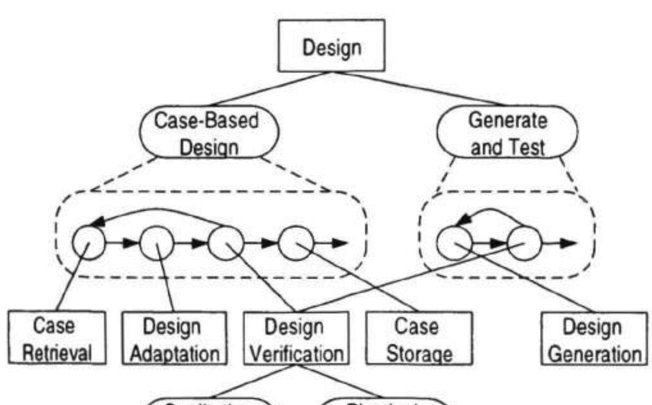

Figure 1: The top few levels of a TMK model of a design process. Rectangular boxes represent tasks; round boxes represent methods. The circle-and-arrow diagrams within the round dotted boxes represent the control portion of the methods.

Consider the top level task in this decomposition: design. The design task might be represented with the following knowledge element:[1]

```
Task design
-domain: physical-devices
-input: (desired-function)
-output: (new-model)
-given: desired-function is feasible
-makes:
  new-model is internally consistent
  AND
  the function of new-model is equivalent
   to desired-function
-by: (case-based-design generate-and-test)
```

This knowledge element asserts that design is a process which, given a desired function, produces a new model, where the knowledge types associated with the desired function and the new-model are defined within the domain of physical-devices.[2] It further asserts that for this task to be performed, the desired function must be feasible and that when the task is completed, the new model should be an internally consistent model whose function is equivalent to the desired function. Lastly, this knowledge element asserts that two methods are known to accomplish this task: case-based design and generate-and-test. Consider the following representation for the case-based design method:

```
Method case-based-design
-domain: physical-devices
-given:
  desired-function is feasible
  AND
  there exists an old-model in the case memory
   such that the function of the old-model
   approximates desired-function
-makes:
  new-model is internally consistent
  AND
  the function of new-model is equivalent
   to desired-function
-subtasks: (case-retrieval design-adaptation
            design-verification case-storage)
-control:
  DO
   (case-retrieval
    design-adaptation
    design-verification)
  UNTIL new-model is verified
  case-storage
```

This method has requirements and results (i.e., `:given` and `:makes`) which are consistent with the design task. However, there is an additional condition required for the method

[1] In this paper, I present examples in English-like pseudo-code rather than provide an elaborate formal exploration of TMK syntax and semantics. A technical presentation of the most recent computational implementation of the TMK language (in the SIRRINE2 agent architecture) appears in (Murdock and Goel, 1998).

[2] The precise content of the domain knowledge element itself is, of course, a major component of the TMK language. In general, domains contain descriptions of both abstract concepts (e.g., the concept of a function) and concrete variables (e.g., the `desired-function` variable). A detailed examination of this portion of the TMK language is, however, beyond the scope of this paper.

to function: there must be a model in the case memory whose function approximates the desired function. The case-based design method sets up four subtasks: retrieval, adaptation, verification, and storage. The control information for the method indicates that the first three subtasks are to be executed in order, repeatedly, until a new model is produced which is verified by the verification subtask, after which the fourth subtask is executed. Under this control, a reasoner based on this model would continue to retrieve, adapt, and verify cases until it either succeeded, or it failed to retrieve any more cases (in which case the requirements for the adaptation task would not be met and the method as a whole would fail).

These examples illustrate the content of the tasks and methods of TMK models. Tasks are defined by (i) the knowledge which they require and produce, and (ii) the methods which accomplish them. Methods are defined by (i) the knowledge which they require and produce, (ii) the subtasks which they establish, and (iii) the ordering requirements that they define for their subtasks.

Topics in Cognition

To explore the notion of TMK as a unifying theme for the study of cognition, let us consider some specific topics and models from the perspective of TMK. In doing so, we will encounter deeper issues regarding the nature of the modeling language.

Visual Pattern Recognition

An extremely basic task which forms a foundation for a wide range of cognition is that of visual pattern recognition. Consider the model of pattern recognition presented in (Treisman, 1988). This model involves four core elements:

- Immediate, simultaneous recognition of certain specific primitive features such as color and size.
- Serial focus of an "attention spotlight" which binds together clusters of features.
- Generation of an "object file" to label such a cluster.
- Recognition of a stored description matching an object file.

This model can be implemented in a TMK model with the following task-method hierarchy:

```
+ Task: Recognize Pattern
  * Method: Feature-Based Pattern Recognition
    + Task: Assign Object File
      * Method: Spotlighting
        + Task: Identify Features
          * Method: Feature Recognition
            + Task: Recognize Color
            + Task: Recognize Orientation
            + Task: Recognize Size
            + Task: Recognize Stereo Distance
        + Task: Focus Spotlight
        + Task: Create Object File
    + Task: Match Stored Description
```

Two particularly interesting issues arise from this representation of this model of pattern recognition. The first of these issues is the representation of the knowledge being accessed; for example, the form of the object file, the nature of the stored object descriptions, and the mechanism for matching the two are complex and interesting problems. I will not consider this issue further because it is largely unspecified in (Treisman, 1988). The other interesting issue raised by this decomposition is the nature of the control information in the methods above. For example, the feature recognition method has four subtasks, all of which are executed simultaneously, as many times as there are inputs available. In contrast, the higher-level spotlighting task is serial and looping in nature; the identify features task runs to completion, then the focus spotlight and create object file subtasks run sequentially, but repeatedly for as many clusters of features as are available.

TMK, in its current form, does not have sufficiently powerful mechanisms for specifying control to implement the ordering requirements described above. TMK does support describing sets of tasks as not being bound by ordering constraints. However, it does not support the execution of multiple instances of the same task simultaneously; TMK could specify a model in which the size, color, orientation, etc. of a single shape were determined in parallel but not one in which these features were identified for multiple shapes at once. The reason for this restriction is that tasks in TMK must be defined as being bound to a specific problem variable (e.g., the desired functionality in the design case); the idea of running a task on all inputs which are currently available does not have a formalization within TMK.

This conflict affords two possible resolutions:

- TMK needs to be enhanced to support running identical tasks in parallel over a range of inputs.
- Visual pattern recognition is outside of the scope of the TMK framework.

The latter resolution is not entirely unreasonable; existing TMK work has largely focused on extremely high-level cognition (such as design) which are generally accepted to be roughly serial in most regards (Newell and Simon, 1972). For such a topic, one could expect to treat pattern recognition as inherently primitive. However, to the extent that we are seeking to use TMK as a unifying framework across levels of abstraction, this is a very unsatisfying solution. I think that it is more fair to say that to the extent to which we want to work at this level, the TMK language needs to be augmented.

Automatization

Automatization is a phenomenon which, while not as primitive as pattern recognition, is still a relatively basic element of cognition. Consider the issue addressed by (Logan, 1988), speed-up learning in simple, repeated tasks such as lexical decision and "alphabet arithmetic" (a more complex task involving determining the truth of equations of the form A+2=C, etc. where the position of letters in the alphabet determines their value). Logan's model could be described with the following task decomposition (for the alphabet arithmetic problem; the lexical decision problem is represented similarly):

```
+ Task: Solve Problem
  * Method: Instance Retrieval
    + Task: Retrieve Instance
  * Method: Analytical Solution
    + Task: Compute First Letter Value
    + Task: Compute Second Letter Value
    + Task: Add numbers
    + Task: Compare
```

Some key commitments of Logan's model which provide particular challenges to the TMK analysis are:

- The instance retrieval and the analytical solution methods are run in parallel and only the results of the first method to succeed are used.
- The time taken by the instance retrieval method when a single instance is in memory should be a variable of a specific distribution (under very broad assumptions).
- The time taken by the instance retrieval method when N instances are in memory should then be the minimum of N such variables.

The TMK language does not specify a mechanism for allowing alternative methods (as opposed to tasks) to be run in parallel; this would, however, be a trivial addition. The second issue is one of how the retrieve instance task is implemented; however, it seems extremely plausible that a reasonable implementation would satisfy this requirement. The third issue is somewhat tricky. We could support this method by performing the retrieve instance task simultaneously on all instances on memory (as per the feature recognition tasks in the previous example). There is a significant difference; the feature recognition task runs in parallel on all inputs and concludes when all inputs are concluded, this task runs on all inputs and concludes when any input is concluded. However, I feel that by augmenting TMK to support both of these sorts of parallelism, we can represent both of these models of cognition.

A key observation here is that immediately prior to solving the problem, the viewers are required to recognize the letters and numbers in the problem. This would be represented in the TMK model as a repeated sequence of pattern recognition tasks. These tasks could, in turn, be represented by the feature based recognition method described in the previous example. In this way, TMK provides a framework for integrating these models.

Implicit Memory

Consider the model of implicit memory tasks from (Jacoby, 1991). This work generally posits two general subtasks of recognition: determination of recollection and determination of familiarity. Two top level tasks are presented to subjects: inclusion recognition (i.e., recognition with priming known to be correct) and exclusion recognition (i.e., recognition with priming known to be incorrect). The methods posited for use in the two tasks both invoke these same two recognition subtasks but they use a different procedure for synthesizing the results. We can describe this model using the following TMK decomposition:

```
+ Task: Inclusion Recognition
  * Method: Inclusion Recognition Method
    + Task: Familiarity Analysis
    + Task: Recollection Analysis
    + Task: Inclusion Result Synthesis
+ Task: Exclusion Recognition
  * Method: Exclusion Recognition Method
    + Task: Familiarity Analysis
    + Task: Recollection Analysis
    + Task: Exclusion Result Synthesis
```

Let us look at some of these items in more detail:

```
Method Exclusion-Recognition-Method
-domain: word-recognition-domain
-makes:
  recognized? holds IF AND ONLY IF
   stimulus was seen in training
   AND
   stimulus was not seen in priming
-control:
   DO IN PARALLEL
    (Familiarity-Analysis
     Recollection-Analysis)
   Exclusion-Result-Synthesis
```

This method states that the exclusion recognition method produces a boolean result `recognized?` which should be true if and only if the word was seen during stimulus training or was seen during stimulus priming. The method has two serial components: first the two memory tasks are executed (in parallel or arbitrary order) and then the results are synthesized. Let us consider the synthesis task:

```
Task Exclusion-Result-Synthesis
-domain: memory-synthesis-domain
-input: (recalled-from-priming? familiar?)
-output: (recognized?)
-makes:
  recognized? holds IF AND ONLY IF
   familiar? AND NOT recalled-from-priming?
-by: exclusion-logic-procedure
```

This specification says that the exclusion result synthesis task takes as input information (derived from the memory subtasks) about whether the stimulus is familiar and whether it is recalled from the earlier (exclusive) priming. It then derives a truth value for whether the word is recognized by a logical inference which holds if and only if the word is familiar and not recalled from the priming.

This analysis shows how the TMK modeling language can encode a memory retrieval mechanism. A great many tasks referred to in cognitive models of other phenomena make use of memory retrieval; it is possible that some of the familiarity and retrieval tasks can be used as subtasks to other, more complex tasks. Using this framework, knowledge derived about these two tasks (e.g., implications of their conditional independence) can be established within this model and then directly applied to other models.

Reflection

If TMK models present a framework by which we, as scientists, can understand cognition, might they also provide a basis for a knowledge account of how humans understand their own cognition? Since humans are able to provide explanations (albeit often incomplete and incorrect ones) of their own reasoning, it is apparent they do have some form of internal knowledge of themselves.

Recall from the introductory section that TMK models are originally derived from early work in the modeling of physical devices. These physical device models have been used for a wide variety of tasks, but the most prevalent task to which they have been applied (and the one to which they are most tightly tuned) is that of adaptive redesign. Consequently, it is not unreasonable to suspect that TMK models might be appropriate to the adaptive redesign of an intelligent reasoner. In fact, some AI research has suggested that TMK models do provide support for such adaptation (Stroulia, 1994; Murdock and Goel, 1998). To the extent that these AI systems use TMK models to redesign themselves, the TMK language can be viewed as encoding the knowledge which enables reflective learning. Because these models have been shown to support this sort of reasoning in artificial agents, it is conceivable that they could approximate the analogous knowledge possessed by humans.

If we accept that TMK might provide a (possibly limited) account of knowledge, there arises the question of whether and to what extent this knowledge is consciously accessible. (Stroulia, 1994, p. 249) argues that reasoning using models of this sort corresponds with *conscious* reflection. Are consciousness and model-based reflection simply different perspectives on the same phenomenon or is the relationship between them more complex?

One obvious measurement of consciousness is introspective accessibility.[3] It is apparent that humans *are* able to introspectively describe their processing mechanisms. However, it is also apparent that humans are severely limited in this ability and frequently produce demonstrably incorrect or incomplete descriptions of their own reasoning. How can we account for this common observation within a cognitive TMK framework? Some possibilities include:

1. TMK models are purely conscious reasoning structures, but they are inherently incomplete and incorrect.
2. TMK models are purely unconscious reasoning structures. To the extent that people can describe them, they are only inferred from the consequences of their use.
3. TMK models are both conscious and unconscious reasoning structures. They are only partially and inaccurately available to conscious thought.

The first possibility seems to be the most superficially obvious choice; given that these models may be elicited from people in an incomplete / incorrect form, the default hypothesis is clearly that people have incomplete / incorrect versions of these models in their conscious memory. However, there are serious problems with this idea; most significantly, much of the work on reflective self-redesign which provides the primary motivation for TMK models as cognitively plausible structures, involves self-adaptations which are difficult to envision being completely accessible to consciousness. I believe that it would be possible, with empirical studies, to show that some adaptive learning scenarios which were typical of the kinds of TMK self-redesign we have proposed involve components of the TMK models which are not consciously accessible. If this were done, we could rule out possibility 1 as an account of the relationship between consciousness and TMK models.

The second possibility also seems initially appealing; since people clearly don't have full access to these models, why should we believe that they have any access at all. It would be difficult to empirically validate or falsify this position; from a behavioral perspective, it is difficult to distinguish between knowledge that exists and knowledge that is inferred whenever it is needed.

However, our knowledge of memory phenomena suggests that it is unlikely that these models are *always* computed, in that many computational tasks have been widely shown to be supplanted by memory access when applicable. For example, both the effect of prior (even "unremembered") exposure to words on fragment completion (Tulving, 1985) and the prevalence of cryptomnesia in more complex tasks (Marsh and Bower, 1993) suggest that, at the very least, specific memory traces of past inference process are available; to the extent that we have had occasion to infer TMK structures for certain reasoning tasks in the past, it is potentially reasonable to claim that these inferred structures are available in memory, even if their source (i.e., the particular reasoning event during which this TMK model was inferred) has been forgotten.

Consider the position taken by (Kahneman and Miller, 1986) with respect to the issue of the existence of norms (i.e., judgements of typical instances of a class) as a memory structure. This view is very similar to the position taken in possibility 3; their paper focuses on post-hoc, inferred norms (analogous to inferred TMK structures) but claim that for some situations, pre-existing, known norms do exist (analogous to remembered TMK structures). Pending further evidence, I would like to make a similar claim here; that, to the extent to which the TMK language provides a plausible account of self-knowledge, this self-knowledge is partially but not fully accessible to conscious memory.

The notion of TMK models as *partially* conscious reasoning structures seems to relate to the notion of a consciousness "fringe" as described in (Mangan, 1993).[4] Can we consider reflective self-representation of the sort embodied in TMK models to be guiding reasoning from the fringe of consciousness? If so, we would expect that attempts to bring such representations into fully conscious focus (for example, in introspective descriptions of one's own reasoning) to resemble other phenomena which involve shifts from fringe to focal consciousness. An example of such a shift which Mangan presents is the "tip-of-the-tongue" (TOT) phenomenon in which people know that they have encountered some fact but can not immediately recall it. To what extent do the limitations and restrictions of conscious access to self-knowledge resemble the limited memory access of TOT? Further research is needed to address these topics.

Conclusions

It is clear that a great many issues need to be resolved before the TMK modeling language can be used effectively as a theoretical framework for synthesizing models of cognition. As I have argued, the TMK mechanisms for specifying control of instantiation of subtasks needs to be enhanced to deal with a variety of different kinds of parallel computation. Furthermore, there do seem to be (at least) two kinds of cognitive issues for which the TMK modeling framework provides little value:

[3]While the limitations of introspection as a general mechanism for studying cognition are well known, it seems like an essential tool for studying consciousness as a phenomenon. It is difficult to argue that some aspect of cognition is conscious but not accessible to introspection; under these assumptions, what would the term "consciousness" mean? For a more detailed look at some of these issues, see (Reisberg, 1997, p. 589, f.f.).

[4]This paper further cites (James, 1890) but presents the notion in the context of a modern, cognitive framework; it is this modern formulation to which I am referring here.

- Cognitive processes which are inherently atomic, i.e., which cannot be further decomposed into elements. These are easily modeled in TMK (i.e., as a single task) but such models provide very little insight. Issues of this sort can be seen as below the level of abstraction for which TMK is useful.
- Cognitive processes which are inherently inamenable to teleological analysis. (van Gelder, 1997) argues that cognition may not be decomposable into causal flows of functional elements. To the extent that this is true of even certain cognitive phenomena, TMK is probably not useful for these phenomena.

Despite these limitations, I believe that TMK models do provide a useful mechanism for integrating models of cognition. The overwhelming majority of models of cognitive phenomena which have been developed do not fall into either of the two categories above: they are complex in nature and are decomposed into functional elements. In providing causal, compositional, functional descriptions of reasoning processes, the TMK language suggests an account of how models of reasoning may be combined to form a deep understanding of cognition. Furthermore, since these models can provide such an understanding, they also form a plausible hypothesis regarding the knowledge content of human self-understanding.

Acknowledgments

I would particularly like to acknowledge the extensive contributions of Jeffrey Toth, both for his presentation and organization of many of the works cited here and for his feedback on earlier drafts of this paper. I would like to further acknowledge the contributions of Ashok Goel, Eleni Stroulia, Dean Allemang, Todd Griffith, Spencer Rugaber, and many others in the development of the TMK language. Portions of this effort were sponsored by the Defense Advanced Research Projects Agency, and the United States Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-96-2-0229. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory, or the U.S. Government.

References

Anderson, J. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22.

Anderson, J. R. (1993). *Rules of the Mind*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Bhatta, S. (1995). *Model-Based Analogy in Innovative Device Design*. PhD dissertation, Georgia Institute of Technology, College of Computing.

Goel, A. K. (1989). *Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving*. PhD dissertation, The Ohio State University, Dept. of Computer and Information Science.

Goel, A. K., Bhatta, S., and Stroulia, E. (1997). Kritik: An early case-based design system. In Maher, M. L. and Pu, P., editors, *Issues and Applications of Case-Based Reasoning to Design*. Lawrence Erlbaum Associates.

Goel, A. K., Gomez, A., Grue, N., Murdock, J. W., Recker, M., and Govindaraj, T. (1996). Explanatory interface in interactive design environments. In Gero, J. S. and Sudweeks, F., editors, *Proceedings of the Fourth International Conference on Artificial Intelligence in Design*, Stanford, California. Kluwer Academic Publishers.

Griffith, T. W. (1997). *A Computational Theory of Dynamical Modeling in Scientific Reasoning*. PhD thesis proposal, Georgia Institute of Technology, Cognitive Science Program, College of Computing.

Jacoby, L. L. (1991). A process-dissociation framework: Seperating automatic from intentional uses of memory. *Journal of Memory and Language*, 30:513–541.

James, W. (1890). *The principles of psychology*. Holt, New York.

Kahneman, D. and Miller, D. T. (1986). Norm theory: Comparing reality to its alternatives. *Psychological Review*, 93:136–153.

Laird, J., Newell, A., and Rosenbloom, P. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychology Review*, 95:492–527.

Mangan, B. (1993). Taking phenomenology seriously: The "Fringe" and its applications for cognitive research. *Consciousness and Cognition*, 2:89–108.

Marsh, R. L. and Bower, G. H. (1993). Eliciting cryptomnesia: Unconscious plagiarism in a puzzle task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 19:673–688.

Murdock, J. W. and Goel, A. K. (1998). A functional modeling architecture for reflective agents. In *Proc. AAAI98 Workshop on Functional Modeling and Teleological Reasoning*, Madison, Wisconsin.

Newell, A. and Simon, H. (1972). *Human problem solving*. Prentice-Hall, Englewood Cliffs, NJ.

Reisberg, D. (1997). *Cognition*. W. W. Norton & Company, New York.

Stroulia, E. (1994). *Failure-Driven Learing as Model-Based Self Redesign*. PhD dissertation, Georgia Institute of Technology, College of Computing.

Treisman, A. (1988). Features and objects in visual processing. *Scientific American*, 254:114–124.

Tulving, E. (1985). How many memory systems are there? *American Psychologist*, 40:385–393.

van Gelder, T. (1997). The dynamical alternative. In Johnson, D. M. and Erneling, C. E., editors, *The future of the cognitive revolution*, pages 227–244. Oxford University Press, New York.