Title
Modeling the Self-explanation Effect with Cascade 3

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 13(0)

Authors
VanLehn, Kurt
Jones, Randolph M.
Chi, M. T. H.

Publication Date
1991

Peer reviewed

Modeling the Self-explanation Effect with Cascade 3

Kurt VanLehn, Randolph M. Jones, & M. T. H. Chi

Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260
VanLehn@cs.pitt.edu, Jones@cs.pitt.edu, Chi@vms.cis.pitt.edu

Abstract

Several investigations have found that students learn more when they explain examples to themselves while studying them. Moreover, they refer less often to the examples while solving problems, and they read less of the example each time they refer to it. These findings, collectively called the self-explanation effect, have been reproduced by our cognitive simulation program, Cascade. Cascade has two kinds of learning. It learns new rules of physics (the task domain used in the human data modeled) by resolving impasses with reasoning based on overly-general, non-domain knowledge. It acquires procedural competence by storing its derivations of problem solutions and using them as analogs to guide its search for solutions to novel problems. This paper discusses several runs of Cascade wherein the strategies for explaining examples is varied and the initial domain knowledge is held constant. These computational experiments demonstrate the computational sufficiency of a strategy-based account for the self-explanation effect.

Introduction

The long-term goal of the Cascade project is to develop a model of knowledge acquisition in scientific task domains. The short-term goal is to model the self-explanation effect. Cascade has been tested using data from a study by Chi, Bassok, Lewis, Reimann and Glaser (1989) who studied students learning classical particle dynamics, the first topic in a typical first-year college physics course. The 8 subjects studied the first three chapters of college textbook, then read the prose part of a chapter on Newton's laws. They took a test on their understanding of the chapter, then studied 3 worked examples and solved 25 problems. Protocols were taken as they studied the examples and solved the problems. On the basis of the scores on problem solving, the 8 subjects were divided two groups. The 4 students with the highest scores were called the Good solvers; the others were called Poor solvers. Since the students in both groups scored the same on pre-tests, the Good solvers seem to have learned more during the experiment (but see discussion below). Using protocol analysis, Chi et al. attempted to find out how the Good solvers managed to learn more than the Poor solvers from the same material. They found five differences:

1. The Good solvers uttered more self-explanations as they studied examples, whereas the Poor solvers' comments were mostly paraphrases of the examples' statements.
2. All students commented frequently on whether they understood what they had just read. The Good solvers tended to say that they did not understand what they had just read, whereas the Poor solvers tended to say that they did understand. Since the Poor solvers scores show that they understood less than the Good solvers, their self-monitoring was less accurate than the Good solvers'.
3. During problem solving, the Poor solvers tended to refer back to the examples more often than the Good solvers.
4. When the Good solvers referred to the examples, they read fewer lines then the Poor solvers. The Poor solvers tended to start at the beginning of the example and read until they found a useful line, whereas the Good solvers started reading in the middle of the example and read only one line.

The first two findings have also been observed in similar studies of students learning Lisp (Pirolli & Bielaczyc, 1989) and electrodynamics (Fergusson-Hessler & de Jong, 1990). This cluster of findings is called the self-explanation effect.

Hypothesized sources of the self-explanation effect

Cascade is based on the hypothesis that the self-explanation effect is caused by knowledge-level learning that occurs as the students explain examples and solve problems. This section introduces the hypothesis by first examining competing hypotheses.

A plausible hypothesis is that the two groups of students had accumulated different knowledge of physics just prior to studying the examples. This difference might be due to prior exposure to physics or to reading the text of the chapter more carefully. The ones who had more prior knowledge solved more problems correctly, and thus were classified as Good solvers. Under this prior-knowledge hypothesis, all subjects try to explain the text and the example lines, but those who already know a lot are better able to explain the example and so produce more self-explanations (finding 1 in the list above). Moreover, because they produce more derivations during example processing, they use fewer (finding 3) and more economical (finding 4) references during analogical problem solving. Thus, the prior-knowledge hypothesis is consistent with 3 of the 4 findings. There are, however, three sets of evidence against the prior knowledge hypothesis.

1. After reading the text of the target chapter, the students in the Chi et al. study took a test on their knowledge of Newton's laws. The mean scores of the Good and Poor students on this test were exactly the same (Chi et al., 1989). This suggest that

both groups of students had roughly the same prior knowledge.

2. Chi and VanLehn (1991) conducted a fine-grained analysis of all 258 self-explanations in the protocols, reducing each to a set of propositions. For each proposition, they attempted to determine whether it was inferred (a) from the example line, (b) from common sense knowledge, (c) from knowledge acquired from previous example lines, or (d) from the text. The first three categories are considered non-text sources. Of the propositions whose source could be determined, 68.5% were inferred from non-text sources. More importantly for the present argument, this proportion was the same for both Good and Poor students. If the Good students had more prior knowledge, more of their propositions would be encoded as coming from the text. Thus, this result is inconsistent with the prior knowledge hypothesis.
3. The prior-knowledge hypothesis predicts that Poor subjects would utter more negative self-monitoring statements because they more often fail to explain a line. In fact, they utter fewer negative self-monitoring statements.

Although it is unlikely that all 9 students had exactly the same prior knowledge, the above difficulties indicate that variations in prior knowledge cannot be the sole source of the self-explanation effect. There must be some kind of learning going on.

Because the subjects are explaining examples, a plausible type of learning is EBL (Mitchell, Keller & Smadar-Cabelli, 1986). EBL is symbol-level learning (Dietterich, 1986), in that all the knowledge is assumed to be present in some form before the learning begins. Learning consists of making the knowledge more efficiently usable. EBL, knowledge compilation (Anderson, 1983) and chunking (Newell, 1990) are all instances of symbol-level learning. However, the hypothesis that self-explanation is caused by symbol-level learning has two difficulties.

1. When the subjects took an untimed test on the content of the text, their mean score was only 5.5 out of a possible 12. Moreover, after studying the examples and solving the problems, the Good student's score increased to 8.5. This suggests that students did not learn much physics from studying the text, and that some more physics was learned by studying the examples and working the problems.
2. The text does not contain all the information needed by the subjects to explain the examples or solve the problems. To quantify this inadequacy, an extensive task analysis and simulation conducted with the aid of Bernadette Kowalski and William Ball. Starting with the task analyses of Bundy, Byrd, Luger, Mellish and Palmer (1979) and Larkin (1983), we developed a set of rules and a representation of physics problems that was simple and yet sufficient for solving all but 2 of the 25 problems in the Chi study. (Solving the 2 problems would require a type of mathematical reasoning that we did not bother to implement). During this time, extensive, albeit informal, analyses of the Chi protocols were conducted in an effort to align the proposed knowledge representations with the subject's comments. The resulting target knowledge base contained 62 physics rules. Next, two people who were not involved in the development of the target knowledge base were asked to judge each rule and determine whether it was mentioned anywhere in the textbook prior to the examples. There was 95% agreement between the judges. Disagreements were settled by a third judge. Of the 62 rules in the target knowledge base, 29 (47%) were judged to be present in the text prior to studying the examples. Thus, more than half the knowledge required for explaining the examples and solving the problems is not presented in the text, and presumably is not known by the subjects prior to explaining the examples and solving the problems.

These results suggest that the major prerequisite of symbol-level learning is not met, for the students did not seem to have complete knowledge before the example studying and problem solving began. Thus, some kind of knowledge-level learning must be going on during the explanation of examples and the solving of problems.

Because the examples contain more information than the problems, a plausible hypothesis is that all knowledge-level learning occurs during the explanation of examples. Using the 33 rules that did not occur in the text, we estimated that only 11 of the rules were used during the examples. The other 22 were first used during the problems. This suggests that two-thirds of the rules are acquired during problem solving. Thus, it appears that some kind of knowledge-level learning is going on during both example explaining and problem solving. This is the hypothesis upon which Cascade is based.

The hardest technical challenge is to find a knowledge-level learning method that can learn correct rules during problem solving. Learning during problem solving is harder than learning during example explaining, because the examples provide partial description of their solution paths which allows the program to do less guessing than it must do when learning during problem solving. Because students often refer to examples as they solve problems, we assume that they are using the example's solutions to constrain their generation of the problems' solutions, and this in turn facilitates correct learning during problem solving. We hypothesize that knowledge-level learning takes place in the context of analogical problem solving and example explaining.

Since the Poor students learn fewer rules during analogical problem solving than the Good students, and the Poor students generated fewer self-explanations during example studying, we hypothesize that there is something about explaining the examples that causes analogical problem solving to be more effective. It generates solution paths that are more often correct and this in turn establishes a better context for knowledge-level learning. We built Cascade order to work out the interactions between example explaining, analogical problem solving and learning, and thus provide a test of the computational and empirical sufficiency of our hypotheses.

The Cascade model

Cascade models two basic activities: explaining examples and solving problems. Knowledge-level learning goes on during both. Because the type of physics problems used in Chi et al.'s study involve only monotonic reasoning in a single state, Cascade uses a rule-based, backchaining theorem prover (similar to Prolog) to implement both activities. A physics example is presented to Cascade as a set of facts representing the givens of the problem and a list of propositions representing the force diagram and the lines of the problem's solution. Cascade explains each proposition by proving that it follows from the givens and the preceding propositions. To solve a problem, Cascade is presented with facts representing the problem's givens and is asked to prove a proposition involving the quantities the problem seeks. For instance, the translation of, "What is the tension of string A?" is "Prove the proposition value(tension(stringA), X)." In the process of proving the proposition, Cascade derives a value for the variable X, thus solving the problem. Although this model of problem solving and example explaining is clearly too simple to cover all task domains, it suffices for physics and other task domains dominated by monotonic reasoning.

Cascade includes two kinds of analogical problem solving. One kind of analogy is used when Cascade has multiple rules that can be applied to achieve a goal and it does not know which one to choose. To get advice, it refers to its derivation of the examples' lines, which are stored as the examples are explained. (A derivation is represented by a set of pairs, each containing a goal from the proof of a line and the rule used to achieve that goal.) This type of analogy begins by retrieving an example (retrieval is currently not modeled in a psychologically plausible way), establishing a mapping between the example's givens and the current problem's givens, then using the mapping to see if the example's derivation has a goal that is equivalent to the goal that it is currently worried about. If it finds an equivalent goal, the rule that achieves that goal is chosen for attempting to prove the worrisome goal. This type of analogy is called analogical search control, because it uses the example as a source of advice on which of several alternative to try first. For instance, a student might say, "I cannot tell whether I should project this onto the x-axis or the y-axis. At an analogous point in the example, they projected onto the x-axis, so I'll try that too." Analogical search control is also used in the Eureka (Jones, 1989).

The second type of analogy is used when Cascade cannot find a rule that will apply to the current goal. Here, it tries to find a line in an old example that it can convert into an appropriate rule. It begins just like analogical search control by retrieving an example and forming a mapping between the givens of the example and the problem. Next it now looks for a line in the example's solution that mentions the current goal (or rather, a goal equivalent to the current goal under the mapping). Most lines are equations, so it is simple to convert a line to a temporary rule which can then be used to try to achieve the goal. For instance, a student might say, "I need some way to get the tension of string A. The example has a line saying that string 1's tension is mg sin 30. Those two strings are analogous, and 30 degrees is analogous to 45 degrees in this problem, so I bet that the tension of string A is mg sin 45 degrees." This type of analogy is called transformational analogy, after a similar method explored by Carbonell (1986). As Carbonell discovered, transformational analogies often yield wrong answers.

Cascade's mechanism for learning at the knowledge level is called explanation-based learning of correctness or EBLC (VanLehn, Ball & Kowalski, 1990). The basic idea is divide knowledge into domain knowledge and non-domain knowledge. Domain knowledge contains rules that are believed to be correct and appropriate for the task domain. Non-domain knowledge contains rules that are believed to be incorrect or relevant only to other task domains. The most important non-domain rules for learning are overly general rules. The basic process of EBLC is to use overly general rules whenever domain rules fail, then to save the particular usage of that rule if its use turns out to be correct. EBLC begins when Cascade reaches a knowledge-level impasse. A knowledge-level impasse occurs when there is no domain rule for achieving a goal and there is no successful alternative solution path that uses only domain rules (VanLehn & Jones, in press, describe how such impasses are detected). To resolve a knowledge-level impasse, Cascade tries to use non-domain rules, such as, "If an object has a part, then the property values of the part and the whole are the same." If the use of such non-domain rules ultimately leads to a successful explanation of an example line or a successful solution to a problem, then Cascade forms a new domain rule that is a specialization of the overly general one. The specialization is chosen so that it is also a generalization of the particular usage. For instance, on one problem Cascade could not determine the pressure in a part of a container even though it knew the pressure in the whole. Since there was no alternative solution to the problem, Cascade was at a knowledge-level impasse. It used the overly-general rule just mentioned, which ultimately led to a solution of the problem. Cascade then formed a new domain rule, "If a container has a part then the pressure in the part is equal to the pressure in the whole."

From a machine learning point of view, Cascade does both knowledge-level learning (via EBLC) and symbol-level learning (via analogical search control). Analogical search control is a form of learning, because each time an example is explained or a problem is solved, the system gains another derivation that can be used by analogical search control. However, analogical search control does not change the set of problems solvable with infinite resources. It only changes the efficiency of the search. Thus, it is a symbol-level learning mechanism.

Cascade's learning is similar to those proposed by existing theories of skill acquisition. We believe that analogical search control can eventually provide an account for the practice effects usually explained by chunking (Newell, 1990), knowledge compilation (Anderson, 1983), and other also symbol-level learning mechanisms. EBLC is similar to proposals by Schank (1986), Lewis (1988), Anderson (1990) and others, which also acquire new knowledge by specializing existing, overly general knowledge. Although all these models of skill acquisition are similar in spirit, they differ in significant ways. For more on the Cascade system and a detailed comparison with its predecessors, see VanLehn and Jones (in press).

Modeling the self-explanation effect with Cascade

Given the learning mechanism of Cascade, a simple hypothesis for explaining the difference between Good and Poor solvers is that Good solvers chose to explain more example lines than Poor solvers. To test this, several simulation runs were made, varying the number of example lines explained and turning on and off various learning mechanisms. All these simulations began with the same initial knowledge state. The initial domain knowledge consisted of the 27 rules that three judges found to be present in the text (see above). The rest of the initial knowledge base consists of 45 non-domain rules, of which 28 represented common sense physics (e.g., a taught rope tied to a object pulls on it) and 17 represented over-generalizations, such as "If there is a push or a pull on an object at a certain angle, then there is a force on the object at the same angle." See VanLehn, Jones and Chi (in press) for a list of the 45 non-domain rules.

The simulation runs

Run 1 was intended to simulate a very good student who explains every line of every example. Cascade first explained the 3 examples in the study, then it solved the 23 problems. (The 2 problems that are not solvable by the target knowledge were excluded.) It was able to correctly solve all the problems. It acquired 22 rules: 7 while explaining examples and 15 while solving problems. The new rules are correct physics knowledge, allowing for the simplicity of the knowledge representation. Moreover, they seem to have the right degree of generality in that none were applied incorrectly and none were inapplicable when they should have been. However, some of the rules dealt with situations that only occurred once in this problem set, so they were never used after their acquisition.

Run 2 was intended to simulate a very poor student who explains none of the example lines. To simulate a student who merely reads an example without explaining it, the lines from the 3 examples were placed in Cascade's memory without explaining them. Thus, there was no opportunity for EBLC to learn new rules nor were any derivations left behind to act as search control for later problem solving. Cascade was given the same 23 problems given to it in run 1. It correctly solved 9 problems. Apparently these problems require only knowledge that Cascade had been given initially. As it solved these problems, Cascade learned 3 correct rules via EBLC. On 6 other problems, Cascade found an incorrect solution. EBLC did not occur on these problems. On the remaining 8 problems, Cascade failed to find any solution or its search went on for so long that it was cut it off after 20 minutes. Although EBLC was used extensively on these problems problems, the rules produced were always incorrect. On the assumption that a poor student would not believe a rule unless it led to a correct solution to a problem, rules acquired during failed solution attempts were deleted.

Run 3 was intended to separate the benefits of EBLC from the benefits of analogy. Cascade studied the examples as in run 1, learning the same 7 rules as on run 1. During problem solving, both analogical search control and transformational analogy were disabled. As would be expected of a symbol-level learner whose learning was turned off, Cascade was slower on run 3 than on run 1 (249 seconds per correctly solved problem vs. 154 seconds for run 1), and it answered only 19 of 23 problems correctly. More importantly, a large interaction was found with EBLC. When analogy is not used during problem solving, EBLC learned 10 rules, only 6 of which were correct. Moreover, three of the 6 were the same three that it learned on run 2. Thus, of the 15 rules learned during problem solving on run 1, 3 can be learned without benefit of the rules learned during example studying, 3 others require the example studying rules but can be learned without analogy, and the remaining 9 require both analogy and the example-studying rules. This finding makes sense. Analogical search control and, to a lesser extent, transformational analogy influence the exact location of impasses, which in turn determine the rules learned by EBLC. Their influence is strong enough that analogy is necessary for EBLC to learn 9 of the 15 rules (60%) acquired during run 1's problem solving.

In order to determine whether this effect is due to transformational analogy or analogical search control, a fourth run was conducted that was similar to run 3 except that only analogical search control was disabled. Cascade still used transformational analogy. This allowed it to get two more problems correct, raising its score to 21 of 23 problems. More importantly, EBLC acquired the same 6 correct rules as on run 3. The fact that no further correct rules were acquired implies that it is analogical search control and not transformational analogy that helped EBLC during run 1. Thus, it appears that analogical search control (or some other kind of search control) is necessary during problem solving if EBLC is to learn successfully.

Explaining the self-explanation correlations

Cascade should be able to explain the four differences observed by Chi et al. (1989) between Good and Poor solvers. Assuming that the number of self-explanatory utterances is directly proportional to the number of lines explained during example studying, the job facing Cascade is to explain why explaining more lines causes better scores on quantitative posttests (finding 1), more accurate self-monitoring (finding 2) and more frequent (finding 3) and more economical reference to the examples (finding 4).

The contrast between runs 1 and 2 indicates that Cascade can reproduce the positive correlation between the number of example lines explained and the number of problems solved correctly. On run 1, it explained all the example lines and got all 23 problems correct; on run 2, it explained none of the example lines and got 9 of the problems correct. Knowing the operation of Cascade, it is clear that having it explain an intermediate number of lines would cause it to correctly answer an intermediate number of problems. So the two extreme points (runs 1 and 2) plus Cascade's deterministic design are sufficient to demonstrate the main finding of the self-explanation effect.

Several mechanisms contributed to this result, and each will be examined in turn. First, when more lines are explained, Cascade is more likely to stumble across a gap in its domain knowledge. Such missing knowledge causes impasses, which lead to EBLC and the acquisition of new rules during example explaining. Of the 19 rules that were learned during run 1 and not run 2, 7 (37%) were learned while explaining examples. As the domain knowledge becomes more complete, performance on problem solving rises. Thus, the more self-explanation, the more EBLC during example studying, and hence the more improvement in problem solving.

The acquisition of rules during example studying helps produce contexts during problem solving that allow EBLC to learn more rules during problem solving even without the aid of analogical search control. Of the 19 rules, run 3 shows that 3 (16%) were acquired in this fashion. These new rules also contributed to the improvement in problem solving.

Analogical search control contributes to the correlation both directly and indirectly. When more lines are explained, more derivations available for analogical search control. Because analogical search control prevents Cascade from going down some dead ends, it directly helps raise the score during problem solving (compare runs 1 and 4). There is an indirect effect as well. Analogical search control causes impasses to occur at places where knowledge is truly missing, rather than at local dead ends in the search space, so EBLC is more often applied to appropriate impasses, and thus more often generates correct domain rules. The remaining 9 of the 19 rules (47%) require analogical search control for their acquisition.

Cascade provides a simple explanation of the correlation between the amount of self-explanation and the accuracy of self-monitoring statements, assuming that negative self-monitoring statements (e.g., "I don't understand that") correspond to impasses, and that positive self-monitoring statements (e.g., "Ok, got that.") occur with some probability during any non-impasse situation. When more example lines are explained, there are more impasses, and hence the proportion of negative self-monitoring statements will be higher. In the extreme case of run 2, where no example lines are explained, all the self-monitoring statements during example processing would be positive, which is not far off from Chi et al.'s observation that 85% of the Poor solver's self-monitoring statements were positive.

The third finding involves the frequency of analogical references during problem solving. Chi et al. observed that during problem solving, the Good solvers make fewer references to the examples than the Poor solvers (2.7 references per problem vs. 6.7). These were mostly physical references, wherein the solver turned to the example and reread part of it. Cascade does not distinguish memory references from physical references. However, it does have two different kinds of analogical references. Analogical search control searches for a sought quantity in the derivation of a solution, while transformational analogy reads consecutive lines in an example looking for one that contains the sought quantity. Suppose we assume that all of the transformational analogy references are physical and that few, say P, of the references due to analogical search control are physical. On the Good solver run, Cascade made 551 references for analogical search control and 40 for transformational analogy. Using the assumption, this would yield $551 \times P + 40$ physical references. On the Poor solver run, Cascade could not use analogical search control because no derivations were available from explaining examples. However, it made 91 references for transformational analogy. If $P < .092$, then $551 \times P + 40 < 91$ and Cascade correctly predicts that the Good solvers make fewer physical references than Poor solvers.

Chi et al. observed that the Good solvers read fewer lines when they referred to examples than the Poor solvers (1.6 lines per reference vs. 13.0 lines per reference). Cascade can model this effect, although an assumption is again needed about the percentage of analogical search control references that are physical. Suppose we assume as before that P of the analogical search control references are physics, and furthermore, assume that a physical reference by analogical search control reads only one line. On the Good solver run, Cascade read 340 lines during transformational analogy and $551 \times P$ lines during analogical search control, for a total of $(551 \times P + 340)/(551 \times P + 40)$ lines per reference. On the Poor solver run, Cascade read 642 lines during its 91 transformational analogy references, for $642/91 = 7.1$ lines per reference. If $P > .017$, then $(551 \times P + 340)/(551 \times P + 40) > 7.1$ and Cascade correctly predicts that the Good solvers read fewer lines per references than the Poor solvers.

Notice that the lower bound (.017) on P does not have to be beneath the upper bound (.092). If P had to be above, say .1 in order to get the lines-per-reference finding correct and below .05 in order to get the reference frequency finding correct, then Cascade could not model both these findings. Thus, these findings jointly have the power to test Cascade, and yet it passed their test.

Discussion

The major technical hurdle in developing Cascade was finding a way to constrain problem solving so that knowledge-level learning could operate correctly during it. This was achieved by adding analogical search control, a form of symbol-level learning. There was no way to tell in advance of running Cascade whether analogical search control was sufficient. Fortunately, it was, and Cascade was able to learn all 15 rules that it needed to learn.

As a model of the self-explanation effect, Cascade is both qualitatively and quantitatively adequate. It exhibits the same qualitative behavior as subjects: It can self-explain examples as well as "paraphrase" them. It can solve problems with and without referring to examples, and its analogical references can both dive into the middle of the example to pick out a single fact (analogical search control) or read the example from the beginning searching for a useful equation (transformational analogy). Cascade's performance is quantitatively similar to the Good and Poor solvers as its reproduction of the 4 self-explanation findings shows.

Cascade is based on the assumption that the self-explanation effect is due solely to a difference in example studying habits rather than a difference is prior knowledge. This is probably too extreme. We plan to explore the tradeoff by fitting protocols of each individual subject. Cascade will be forced to explain exactly the lines that the subject explains. When given problems to solve, Cascade should reach impasses in the same places that the subject does. However, the subject will probably display more impasses than Cascade, thus indicating gaps in the prior knowledge. Thus, we should be able to tell exactly how much variation in performance is due to prior knowledge and how much is due to learning strategies.

As a general model of cognitive skill acquisition, Cascade shows promise but needs considerable work. In order to be a more complete account of the phenomena at hand, it needs a model of analogical retrieval and of the difference between physical and mental references to the examples. We believe the existing mechanisms can also handle some well-known phenomena of skill acquisition, such as practice and transfer effects, but this needs to be demonstrated. The major limitation on the generality of Cascade 3 is its use of monotonic single-state reasoning. With the help of Rolf Ploetzner, we are currently incorporating a version of the situation calculus which will greatly enhance the types of reasoning Cascade can model, and thus the number of task domain that it can model. We are encouraged to extend Cascade to become a more complete, more general model of learning by its similarity to other theories of cognitive skill acquisition (e.g., Anderson, 1990; Schank, 1986). It is considerably simpler than those theories and probably more thoroughly implemented and tested. We hope that its simplicity and empirical adequacy remain intact as it is extended.

Acknowledgements

This research was supported by the Cognitive Sciences division of ONR (N00014-88-K-0086) and the Information Sciences division of ONR (N00014-86-K-0678).

References

Anderson, J.R. (1983) *The Architecture of Cognition*, Cambridge, MA: Harvard.

Anderson, J.R. (1990) *Adaptive Control of Thought.* Hillsdale, NJ: Erlbaum.

Bundy, A., Byrd, L., Luger, G., Mellish, C. & Palmer, M. (1979) Solving mechanics problems using meta-level inference. In *Proceedings of the Sixth IJCAI.* Los Altos, CA: Morgan-Kaufman.

Carbonell, J. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R.S. Michalski, J.G. Carbonell & T.M. Mitchell (Eds.) *Machine Learning, An AI Approach: Vol. 2.* Los Altos, CA: Morgan-Kaufman.

Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P. & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13, 145-182.

Chi, M.T. H. & VanLehn, K. (1991) The content of physics self-explanations. *Journal of the Learning Sciences*, 1(1), 69-106.

Dietterich, T.G. (1986) Learning at the knowledge level. *Machine Learning*, 1, 287-316.

Ferguson-Hessler, M.G.M. & de Jong, T. (1990). Studying physics texts: Differences in study processes between good and poor solvers. *Cognition and Instruction*, 7, 41-54.

Jones, R.M. (1989) *A model of retrieval in problem solving.* PhD Thesis, Information and Computer Science, University of California at Irvine.

Larkin, J. (1983) The role of problem representation in physics. In D. Gentner & A. Collins (Eds.) *Mental Models*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Lewis, C. (1988) Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science*, 12, 211-256.

Mitchell,T.M., Keller, R.M. & Kedar-Cabelli, S.T. (1986) Explanation-based generalization. *Machine Learning*, 1, 47-80.

Newell, A. (1990) *Unified Theories of Cognition*, Cambridge, MA: Harvard.

Pirolli, P. & Bielaczyc, K. (1989). Empirical analyses of self-explanation and transfer in learning to program. In *Proceedings of the 11th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum.

Schank, R.C. (1986) *Explanation Patterns: Understanding Mechanically and Creatively*, Hillsdale, NJ: Erlbaum.

VanLehn, K., Ball, W. & Kowalski, B. (1990). Explanation-based learning of correctness: Towards a model of the self-explanation effect. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum.*

VanLehn, K., Jones, R.M. & Chi, M.T.H. (in press). A model of the self-explanation effect. *Journal of the Learning Sciences*

VanLehn, K. & Jones, R.M. (in press). Integration of analogical search control and explanation- based learning of correctness. In S. Minton & P. Langley (Eds.) *Planning, Scheduling and Learning.* Los Altos, CA: Morgan Kaufman.