Toward A Theoretical Account of Strategy Use and Sense-Making in Mathematics Problem Solving

Hermina J. M. Tabachneck*
Department of Psychology
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
tabachneck@cmu.edu

Kenneth R. Koedinger*
Human-Computer Interaction Inst.
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
koedinger@cmu.edu

Mitchell J. Nathan
Learning Technology Center
Vanderbilt University
Box 45 Peabody
Nashville TN 37203
nathanmj@ctrvax.vanderbilt.edu

Abstract

Much problem solving and learning research in math and science has focused on formal representations. Recently researchers have documented the use of unschooled strategies for solving daily problems -- informal strategies which can be as effective, and sometimes as sophisticated, as school-taught formalisms. Our research focuses on how formal and informal strategies interact in the process of doing and learning mathematics. We found that combining informal and formal strategies is more effective than single strategies. We provide a theoretical account of this *multiple strategy effect* and have begun to formulate this theory in an ACT-R computer model. We show why students may reach common impasses in the use of written algebra, and how subsequent or concurrent use of informal strategies leads to better problem-solving performance. Formal strategies facilitate computation because of their abstract and syntactic nature; however, abstraction can lead to nonsensical interpretations and conceptual errors. Reapplying the formal strategy will not repair such errors; switching to an informal one may. We explain the multiple strategy effect as a complementary relationship between the computational efficiency of formal strategies and the sense-making function of informal strategies.

Introduction

Much research on problem solving and learning in math and science has focused on the formal representations and procedures that are the stuff of traditional textbooks (Anderson, Greeno, Kline, & Neves, 1981; Larkin, McDermott, Simon, Simon, 1980). Our current understanding of how formal, algorithmic approaches operate is well grounded. However, people often rely on their implicit understanding of situations and the behavior of quantities in a situation to solve problems and answer questions. This understanding is a foundation on which a solver, on an unfamiliar or taxing problem, may formulate new problem-solving strategies. The documented use of unschooled strategies for solving daily problems suggests that these strategies can be as sophisticated and as effective as school-taught formalisms (Baranes et al., 1989; Hall et al., 1989; Koedinger & Tabachneck, 1994; Lave, 1984; Scribner, 1984; Stigler & Perry, 1989). While informal mathematics is often sufficient for everyday situations, formal mathematics, with its abstract and syntactic nature, provides real leverage in difficult situations which, though less frequent, are often the most consequential. Formal mathematics is not flawless, however. The abstraction process can lead to nonsensical interpretations and syntactic manipulation can be error prone. Informal strategies can help here to make sense of problem situations. Thus, rather than seeing formal and informal approaches as competing, our research has focused on how formal and informal strategies complement each other in the process of doing and learning mathematics.

We provide preliminary results of a computer modeling effort directed at this issue. Informed by quantitative findings and analyses of verbal protocols, our developing model shows how solvers use their understanding of a problem situation to characterize it mathematically. We show why students can reach impasses with algebra, and how the use of informal strategies can lead to the better problem-solving performances of our experimental subjects. The emerging model of people's use of problem-solving strategies shows solvers to be opportunistic and flexible. The weaknesses of one strategy are often compensated for by combining it with another to aid comprehension, facilitate its mathematization, or to support the computational solution.

The Multiple Strategy Effect

Koedinger and Tabachneck (1994) observed a high frequency of informal, non-algebra strategies (used in 83% of all solution attempts) in college students' solutions of algebra word problems. The subjects, twelve Carnegie Mellon undergraduates, were simply asked to "solve these problems"; algebra was never mentioned. Subjects were trained to give a "think-aloud" concurrent verbal protocol (Ericsson and Simon, 1984) and were audio-taped.

<u>Example Problem:</u> A man has 3 times as many quarters as he has dimes. The value of the quarters is one dollar and 30 cents more than the value of the dimes.
Q1a: How many dimes does the man have? (answer: 2 dimes)
Q1b. If the total value of the coins had been 2.55, how many coins would the man have had altogether? (answer: 12 coins)

Strategy and Representation Identification

We identified four different strategies in subject protocols which we call: algebra, guess-and-test, verbal-math, and diagram (described below). The four strategies make use of the following external representations listed roughly in order from more natural ones to more formal ones: 1) verbal

* The order of first two authors is arbitrary.

To appear in the *Proceedings to the 1994 Annual Meeting of the Cognitive Science Society*, Atlanta, GA.

propositions, 2) verbal arithmetic (arithmetic operations expressed verbally), 3) verbal algebra (equations expressed verbally), 4) diagrams, 5) written arithmetic (arithmetic operations expressed in traditional symbols), and 6) written algebra (equations expressed in traditional symbols). As illustrated in Figure 1, each strategy involves movement between representations, or translations, and manipulations within a representation, or transformations (cf. Lesh, Post, & Behr, 1987). Some representations are associated with a specific strategy, for example, written algebra with the algebra strategy, while other representations are used in more than one strategy, for example, verbal arithmetic is used with guess-and-test and verbal-math.

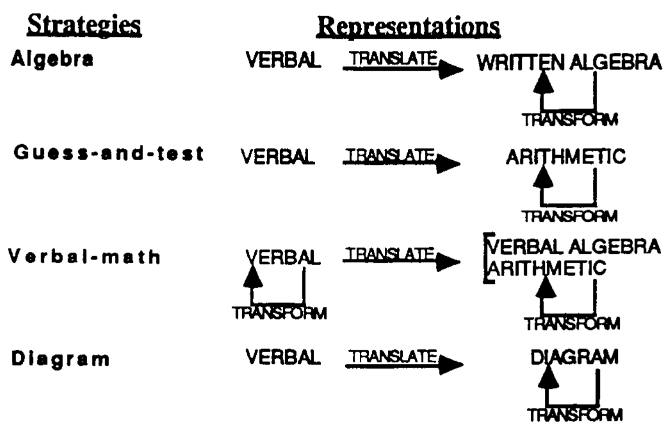

Figure 1

1. Algebra (ALG): The most formal strategy employed by subjects. The verbal problem statement is translated to algebraic assignments and equations. The equations are transformed to find a solution (solve for the unknown).

2. Guess-and-test (G&T): The verbal problem statement is translated into calculation recipes, represented either verbally or as written arithmetic (e.g., "he drove 5 miles more than he biked" is translated into "the miles driven is calculated by adding 5 to the miles biked"). A value for an unknown is guessed at and that value is propagated through the recipes. If a computed value ever conflicts with a given value, then the guess is wrong and new guesses are made. While we call this strategy "guess-and-test", sometimes problem solvers leave off the test phase and use just the guess and propagation phases to develop a better understanding of the problem situation (cf., the model-based strategy in Hall et al., 1989).

3. Verbal-math (VM): The verbal problem statement is transformed into alternative verbal forms. There are two types of transformations: 1) verbal recodings intended to facilitate translation or 2) qualitative operations to estimate unknown values. Included in this strategy are translations to "verbal algebra" where equations are described verbally and transformations are performed that are analogous to written algebra transformations.[1]

[1] Verbal algebra was performed by subjects in one condition of Mayer (1982) and is a generalizations of the "ratio" strategy identified by Hall, et. al. (1989).

4. Diagrammatic (DG): The verbal problem statement is translated into a diagrammatic representation. Transformations are performed on the diagram, including annotations and diagram supported inferences.

Multiple Strategy Use Correlates with Success

The analyses yielded no reliable differences in performance as a function of strategy use. Of the 36 solutions, informal strategies produced correct solutions 65% of the time while the formally taught algebra strategy produced a correct solution 54% of the time — a difference which was not statistically significant. Additionally, solutions used one strategy or more than one strategy. Students were found to be more effective when they used multiple strategies in solving a problem than when they stuck with a single strategy. Of the 19 solutions in which more than one strategy was used, 15 of them or 79% were correct. All 19 solutions involved at least one switch from a formal to an informal strategy (or vice versa). 13 of those involved ALG - VM switches. Of the 17 solutions involving a single strategy, 7 of them or 41% were correct ($X2(36,1) = 4.2$, $p = .02$). In other words, multiple strategy use was about twice as effective as single strategy use. Koedinger and Tabachneck (1994) termed this advantage the Multiple Strategy Effect. Other candidate features of successful performance (e.g., interactions with aptitude) did not distinguish problem solving success from failure.

Impasses, sticking, or switching. The protocols were classified into five stages relating to problem solving:

1. Parsing and Understanding - reading or rereading of problem statement and question, and superficial transforms of the problem statements
2. Solution Setup - setting up the problem solution (e.g., formulating an equation, drawing a diagram, making a guess).
3. Solution - carrying out the computations
4. Solution Answer - finding the answer resulting from the previous computation
5. Problem Answer - finding the answer to the problem question using the solution answer.

Because these problems were challenging to these students, they rarely went through these steps in a single linear sequence. More often students were observed to make *some* progress through these stages, then return to a previous stage, almost always stage 1, and proceed through the stages again (cf. progressive deepening in Newell & Simon, 1972). We used these return events to operationalize a notion of "getting stuck" or reaching an "impasse". Problem solving between impasses is an episode. Each episode was coded for the strategy subjects used in that episode. A strategy does not need to be used to completion in order to be coded as a strategy use.

On some solutions attempts (38%) subjects did not reach an impasse, either because a correct solution was found without trouble or because an error went unnoticed (50% successes). When solution attempts involved an impasse, sticking with the same strategy and trying it again always led to failure (0% successes), while switching strategies was far more likely to be successful (79% successes).

Understanding the Multiple Strategy Effect

Scope of the effect. Multiple strategies will not be effective in routine or easy tasks, as problem solvers can apply a single special-purpose strategy with high success. It only makes sense to use multiple strategies in novel or complex domains where the chance of error is substantial (e.g., as in the 40% error rate in this study). Further, in non-routine tasks, multiple strategies will only be more effective than a single strategy if errors made with the one strategy are not the same errors made with another strategy, that is, if multiple strategies are somewhat independent in their probability of being effective. This will happen only if they have features which complement each other.

Differences in strategy features. Although there are other features that distinguish these strategies, we present the following three features as a minimal set of features for illustrating their complementary strengths and weaknesses: 1) the difficulty of translation or in other words, the "distance" between the representation of the problem situation and the representation used for computing an answer, 2) the efficiency of computing within this representation, and 3) the working memory demands of computing within this representation. See Figure 1 for the connection between the strategies and the representations in which computations are performed. Table 1 shows how the strategies rank on each of these features.

Table 1: Strategy differences

	DISTANCE FROM SITUATION	COMPUTATIONAL EFFICIENCY	WORKING MEMORY DEMANDS
ALG	far	efficient	low
G&T	close	inefficient	low
VM	close	efficient	high
DG	medium	varies	low

*ALG=Algebra, G&T=Guess-and-test, VM=Verbal-math, DG=Diagram

As shown in the first column of Table 1, the *algebra strategy* has the advantage of easing the process of computing answers. This is because the essence of complex problem situations can be abstracted into short strings that can be manipulated by well-defined rules. However, the process of translating problem situations into abstract strings is far from trivial. In addition, the computations performed on these abstract strings are done without connection to the situational context and thus can lead to strings lacking meaningful situational interpretations. These are both a consequence of the distance between the situation and algebra notation (see column 1 of Table 1).

In contrast to algebra, the *guess-and-test* and *verbal-math* strategies involve use of more familiar verbal and arithmetic representations that are closer to the situational context of the problem. Thus, conceptual errors are less likely to occur and more likely to be repaired if they do (cf. Hall et al., 1989). However, *guess-and-test* and *verbal-math* have their own weaknesses. The numerous iterations of guessing that may be required in guess-and-test make it computationally inefficient -- particularly for non-integer solutions. Computation within the verbal-math strategy is analogous to algebra and can be just as efficient. However, unlike algebra where the symbolic strings are short and written down, verbal-algebra involves much longer strings which are not written down and must be maintained in working memory. The difficulty of maintaining such strings means that this strategy is rarely used to find solutions to complex problems. Like guess-and-test, this strategy is often used within multiple strategy solutions to aid comprehension but then is abandoned in favor of algebra for performing the computations.

Like the algebraic strategy, comprehension within the *diagram strategy* consists of attempts to translate the problem statements into an alternative abstract representation. However, this representation results in a structure that maintains some of the semantics of the problem and thus is at an intermediate distance from the problem situation. For instance, in our example problem, one subject drew rows of bigger circles (quarters) and smaller circles (dimes). It is this semantic-preserving quality that makes it easy to understand. Translation and calculation demand little capacity since the structures are drawn on paper in sequential small steps. The computational efficiency varies: problems don't lend themselves equally well for diagramming, and it is not always easy to draw the diagram in such a way that the unknown is easily computable (or even visible).

A Cognitive Model

Our modeling has focused on clarifying one of the ways that multiple strategies can be effective: the use of informal strategies for "sense-making" during problem solving. We model how subjects' informal strategies operate and how the use of these strategies in concert with algebra circumvents impasses and leads to greater problem-solving performance than single strategy use. The current model provides an account of the following:

- *How conceptual errors occur* -- these can be generated by shallow comprehension processes that may fail to distinguish between a quantity and an intensive measure of that quantity or may incorrectly infer logical structure from the syntactic structure.
- *How sense-making strategies can avoid errors.*

The model is written within the ACT-R theory which has a long history of success as a unified theory of cognition (Anderson, 1993).

Modeling how Conceptual Errors Occur

Because written algebra is the most abstract representation, it requires both the largest translation effort and the transformations done to compute answers are done with the least semantic support from the problem situation. The latter fact is particularly relevant when shallow comprehension processes produce algebraic structures that do not make sense or conflict with the problem statement. Because of the abstractness of algebra such conflicts go unrecognized. For example, we observed in the protocols a frequent bug of this sort, the *value-number bug*.

Protocol example. The value-number bug is illustrated in the protocol in appendix 1a. In lines 4-6, the student translates the problem statement "A man has 3 times as many quarters as he has dimes" into "number of dimes is equal to x, and the number of quarters is 3x". Despite his verbal use of "number of dimes" (L5) and "value of dimes" (L10), this distinction is lost in the translation to algebra between lines 10 and 11. Here, "The value of the quarters is one dollar and 30 cents more than the value of the dimes" is translated to "3x = x + 130". This formulation ignores the facts that dimes are 10 cents in value and quarters are 25 cents and thus, the correct equation should be "3*25x = 10x + 130". By assigning 3x to the number of quarters and x to the number of dimes, one strips the units and the associated meaning in going to the abstract representation. This assignment strips both the "coinness" and the "quarter-and-dime-ness" from the representation, and the cue "value" in the second sentence is simply not seen as relevant. After abstracting to 130 from "one dollar and 30 cents", the left and right hand sides of the equation can be put together without ringing conceptual warning bells. The computation of the resulting equation is carried out efficiently and results in a value of x = 65. In line 15, the student states the unit of the result as 65 *cents* presumably because 130 is in cents. In line 16, he appears to realize that 65 cents is an unlikely result for dimes (at this point having conceptually merged the notion of number and value of dimes). The student's response to this impasse is stick with the algebra strategy and try it again, more carefully (not shown). In two more attempts he makes the same error and finally gives up.

Model description. Our model of comprehension and translation involves production rules that look for known patterns in problem statements that can be translated to other representations like algebraic assignments and equations. Terms or phrases encountered in earlier problem statements which refer to the same object or quantity must be properly mapped to the internal representation of that object or quantity. In general, this mapping cannot be done verbatim because problems often use different terms or phrases of the same referent. Also, in capturing the appropriate gist of a problem, specific features are often appropriately ignored such as "The man" in problem 1a. However, this gist-oriented comprehension can sometimes go awry as illustrated by the value-number bug. Below, we illustrate a production rule from our model, in pseudo-English form, that comprehends a noun phrase of the form "... <quantity-term> ..." by mapping to the same working memory element (wme) that the use of "<quantity-term>" in a previous phrase was mapped to.

Comprehend*noun-phrase*shallow-mapping
IF goal is to comprehend the noun phrase "...<quantity-term>...",
  and <quantity-term> was represented before as <quantity-wme>
THEN
  represent "... <quantity-term> ..." as <quantity-wme>.

In the case of problem 1a, this production inappropriately strips away relevant information as illustrated in this example application of the rule:

  The goal is to comprehend "value of the quarters",
  and "quarters" has already been represented as QUARTERS[2]
THEREFORE
  represent "value of the quarters" as QUARTERS

The second column in Appendix 1a provides an abstracted trace of our model on problem 1a parameterized to perform only the algebra strategy. Like the subject, the model ignores the number-value distinction. It applies the *comprehend*noun-phrase*shallow-mapping* production in cycles 7 and 8 (for both dimes and quarters). The consequence of this shallow-mapping is shown in cycles 11 and 12 where the model maps QUARTERS and DIMES back to the algebraic representation defined in cycles 2-6. In cycle 15, the model reaches an impasse as it recognizes that the result, 0.65, does not make sense as a number of dimes.

Modeling how Sense-Making Strategies Avoid Errors

The value-number bug did not appear in either the guess-and-test or verbal-math strategies in any of the protocols. Because these strategies remain in the verbal propositional representation, the meanings of quantities and their role in the problem (e.g., the number of dimes) are maintained.

Protocol example. Like subject 3, subject 9 also started with algebra and made the value-number bug. However, after recognizing that the answer she had found did not make sense, this subject switched to the guess-and-test strategy. The first column of Appendix 1b picks up the protocol at this point. After making a guess at the number of dimes being two (line 17), she immediately thinks of these dimes as totaling 20 cents (line 18) and thus avoids the value-number bug. Using the constraint, identified earlier, that the quarters is 130 cents more than the dimes, she determines the value of quarters to be "a dollar 50" (line 19). She repeats or rehearses this reasoning in lines 20-22. In line 23, she reasons that "a dollar 50" is 6 quarters (doing an embedded guess-and-test). Finally, she returns to the constraint in the first sentence of the problem to determine that if there were 2 dimes there should be 6 quarters. This is consistent with the previous result, so the guess is accepted as the solution (which is correct).

Model description. As with the algebra strategy, we assume that subjects using the guess-and-test strategy may comprehend problem statements equally shallowly and initially be no more aware of the value-number distinction. Thus, the *comprehend*noun-phrase*shallow-mapping* production shown above is also applied here (cycle 28 in the model trace). However, when subjects begin to guess values and propagate them through the problem constraints, the need to convert from number of coins to the value is supported by the situational context that is still active. This is modeled with the following production:

G&T*apply-addition-recipe*convert-input-unit
IF the goal is to apply the arithmetic recipe:
  <output-quantity> can be found by adding <known-quantity>
    to <guessed-quantity>,
  and <known-quantity> is known to be <value1> <unit1>,
  and <guessed-quantity> is guessed to be <value2> <unit2>,

[2]Here QUARTERS indicates a working memory element.

and <unit1> is not the same as <unit2>,
and one <unit2> is <convert-factor> <unit1>'s
THEN
<value2> <guessed-quant> is <convert-factor>*<value2> <unit1>

In trying to compute the quarters, this production recognizes that to add 2 dimes to 1.30 dollars, the dimes most first be converted to 0.20 dollars. In this case, the production would be instantiated as follows:

the goal is to apply the recipe:
quarters can be found by adding one dollar and 30 cents to the dimes,
and one dollar and 30 cents is known to be 1.30 dollars,
and dimes is guessed to be 2 dimes,
and dollars is not the same as dimes,
and one dime is 0.10 dollars,
THEREFORE
2 dimes is 0.20 dollars

This is shown in cycle 30. At cycle 31, the model finishes applying this recipe. It then needs to convert the output of the recipe which is in dollars back to quarters (line 32) and finally check that applying the first recipe (quarters is 3 times dimes) yields the same result.

Summary. Clearly, individual strategies have inherent weaknesses. Our model illustrates how switching representations can make problem solving more successful by compensating for these weaknesses. Some unschooled strategies (e.g., VM and G&T) naturally support comprehension by retaining the problem semantics, thus avoiding common conceptual errors. In the multiple strategy example provided, the strategies were used non-interactively. Other protocols show how strategies can be used interactively as results from an intermediate strategy feed back into the original one. For example, subjects may start with the algebra strategy, run into an error, turn to another strategy to aid the comprehension and translation subtasks, and go back to algebra to solve the problem. Multiple strategy use allows one to adapt to the local difficulties of a problem through selection of a strategy that is best for each subprocess.

Conclusion

Problem solvers regularly use unschooled strategies. Since integrating such common sense strategies and formal mathematics has met with instructional success (Carpenter & Fennema, 1992; Cobb et al., 1991), exactly how these strategies function and interact with previously learned problem-solving strategies is a matter of great interest to cognitive scientists and educators. We showed through a computational model how multiple strategy use helps solvers to understand a problem and to compensate for the weaknesses of a given strategy and their own limitations in executing it. Our ACT-R model is intended to capture the cognitive processes involved in selecting, performing, and integrating multiple strategy use in algebra problem solving. The emphasis here is not on the identification of strategic alternatives to algebra (though the VM strategy for verbal algebra is somewhat novel) -- others have identified similar strategies (e.g., Baranes et al, 1989; Hall et al., 1989).

The emphasis is on understanding how these strategies are used interactively, one strategy providing a sense-making function when another failed. To understand this compensatory interaction, we have identified dimensions of strengths and weaknesses of these strategies: representational format, familiarity, computational efficiency, and capacity demands. A major focus is to understand, through modeling, the specific circumstances under which each strategy is effective both as a basic research goal and for pedagogical purposes. Further work will identify issues in strategy development intended to provide a theoretical explanation of how more familiar student-developed representations and strategies (e.g., guess-and-check, verbal algebra) can be used as ramp in the acquisition of unfamiliar formal representations and strategies (written algebra) (Kaput, 1989).

Formal strategies facilitate computation because of their abstract and syntactic nature, however, the abstraction process can lead to nonsensical interpretations and syntactic errors. The remedy to preventing or repairing such errors is not more careful reapplication of the formal strategy, rather problem solvers are better off switching to an informal strategy to make sense of the problem situation or to identify conceptually inconsistent slips. We explained the multiple strategy effect in terms of the complementary relationship between the computational efficiency of formal strategies and the sense-making function of informal strategies. Effective problem solving in novel situations results from opportunistic and flexible application of both formal and informal strategies.

Acknowledgement

The first author was supported by the James S. McDonnell Foundation grant #92-5.

References

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

Anderson, J. R., Greeno, J. G., Kline, P. J., & Neves, D. M. (1981). Acquisition of problem-solving skill. In J. R. Anderson (Ed.), Cognitive Skills and their Acquisition. Hillsdale, NJ: Erlbaum.

Baranes, R., Perry, M., and Stigler, J. W. (1989). Activation of real-world knowledge in the solution of word problems. *Cognition and Instruction*, 6 287-318.

Carpenter, T. & Fennema, E. (1992). Cognitively Guided Instruction: Building on the knowledge of students and teachers. *International Journal of Educational Research, 17*, 457-470.

Cobb, P., Wood, T., Yakel, E., Nicholls, J. , Wheatley, G., Trigatti, B., & Perlwitz, M. (1991). Assessment of a problem-centered 2nd-grade mathematics project. *Journal for Research in Mathematical Education, 22*, 3-29

Ericsson, K.A. & Simon, H.A. (1984). Protocol Analysis: Verbal reports as data. Cambridge, MA: MIT Press.

Hall, R., Kibler, D., Wenger, E. and Truxaw, C. (1989). Exploring the episodic structure of algebra story problem solving. *Cognition and Instruction, 6* (3).

Kaput, J.J. (1989). Linking representations in the symbol systems of algebra. In S. Wagner & C. Kieran (Eds.),

Research issues in the learning and teaching of algebra (167-181). Reston, VA: NCTM.
Koedinger, K.R., & Tabachneck, H.J.M. (1994). *Two strategies are better than one: multiple strategy use in word problem solving*. Annual meeting of the American Educational Research Association, New Orleans, LA.
Larkin, J., McDermott, J., Simon, D., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. *Science, 208*, 1335-1342.
Lave, J. (1984). In B. Rogoff & J. Lave (Eds.), *Everyday Cognition*. Cambridge, MA: Harvard University Press.
Lesh, R., Post, T., & Behr, M. (1987) Representations and translations among representations in mathematics. In C. Janvier (Ed), *Problems of representation in the teaching and learning of mathematics*, (pp. 33-40). Hillsdale: Lawrence Erlbaum Associates.
Mayer, R. E. (1982). Different problem-solving strategies for algebra word and equation problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8*, 448-462.
Newell, A. and Simon, H.A. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.Inc.
Scribner, S. (1984). Studying working intelligence. In B. Rogoff & J. Lave (Eds.), *Everyday Cognition* (pp. 9-40). Cambridge, MA: Harvard University Press.
Stigler, J. W. & Perry, M. (1990). Mathematics learning in Japanese, Chinese, and American Classrooms. In J. W. Stigler, R. A. Schweder, and G. Herdt (Eds.) *Cultural Psychology*. New York: Cambridge University Press.

APPENDIX 1: DATA-MODEL MATCH

1a line:	**Verbal report, Subject 3, Problem 1a**	cycle:	**Model output of buggy algebra version, Problem 1a**
1-3	[Reads problem statement]	1	Ah, an algebra problem.
4-6	So, we're going to say dimes are equal to x ... number of dimes is equal to x and the number of quarters is 3x	2-6	X = DIMES 3X = QUARTERS
7-9	[repeats 3-6]		[Reading 'value-of DIMES' as 'DIMES' and 'value-of QUARTERS' as 'QUARTERS']
10	and the value of the quarters is one dollar and 30 cents more than the value of the dimes	7-9	QUARTERS is DIMES plus ONE-DOLLAR-AND-30-CENTS.
		10	They want to know the number of DIMES.
11	so, 3x = x + 130	11-12	3X = X + 1.30
12	so then I subtract x from the one side,		
13	I get 2x=130		
14	x= 2/130 which is ... uh		
15	6 ... 65 cents,	13-14	X = 0.65
16	which does not make sense.	15	Solution value 0.65 does not make sense. DIMES should be an integer
17-33	[Repeats algebra and error twice more]		[Note: Like the subject, the model will
34	I get x = 65.		continue to retry algebra and make the same
35	So I have a value ... it's not making sense, though.		error.]

1b line:	**Verbal report, Subject 9, Problem 1a**	cycle:	**Model output of buggy-algebra + guess-and-test version, Problem 1a**
1-14	[Buggy algebra translation. Similar to 1a above.]	1-16	[Buggy algebra translation. Same as 1a above.]
15-16	[Rereads problem indicating impasse and switches to the Guess-and-test strategy]	17-18	I'm going to try the guess-and-test method.
		19-28	[Recomprehending problem sentences.] QUARTERS is DIMES plus ONE-DOLLAR-AND-30-CENTS.
17	right ... so if he had ... if he had 2 dimes,	29	Guessing value for DIMES is 2.
18	that would be 20 cents,	30	2 DIMES would be 0.2 DOLLARS
19	and if he had a dollar 50 in quarters,	31	Because DIMES-IN-DOLLARS is 0.2 DOLLARS and ONE-DOLLAR-AND-30-CENTS is 1.3 DOLLARS, QUARTERS-IN-DOLLARS is 1.5 DOLLARS.
20-21	how many would that be? That would be ... well, if he had two dimes, that would be 20 cents.		[Repeat of cycles 29-30]
22	and then, 20 + 130 would be 150.		[Repeat of cycle 31]
23	and that would be ... how many quarters would that be? ... that would be 4, 5, 6 ... that would be 6 quarters	32	1.5 DOLLARS would be 6.0 QUARTERS
24	2 times 3 is 6 ... oh, ok. Easy.	33-35	Because DIMES is 2 and 3 is 3, QUARTERS is 6.0. That's right! That's what I got before.
25	Uh ... the man has 2 dimes.	36	The problem was asking for DIMES. DIMES was found to be 2.