DTIC FILE COPY.

AD-A218 927

Tower-noticing triggers strategy-change
in the Tower of Hanoi:
A Soar model

Technical Report AIP - 66

Dirk Ruiz and Allen Newell

Departments of Psychology and Computer Science
Carnegie Mellon University

The Artificial Intelligence
and Psychology Project

DTIC
ELECTE
MAR 12 1990
D

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

90 03 12 026

Tower-noticing triggers strategy-change
in the Tower of Hanoi:
A Soar model

Dirk Ruiz and Allen Newell

Departments of Psychology and Computer Science
Carnegie Mellon University

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
Unclassified	

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT
	Approved for public release;
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	Distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
AIP — 66	

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
Carnegie-Mellon University		Computer Sciences Division Office of Naval Research

6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
Department of Psychology Pittsburgh, Pennsylvania 15213	800 N. Quincy Street Arlington, Virginia 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
Same as Monitoring Organization		N00014-86-K-0678

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS p4000ub201/7-4-86			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
	N/A	N/A	N/A	N/A

11. TITLE (Include Security Classification)

Tower-noticing triggers strategy-change in the Tower of Hanoi: A Soar model

12. PERSONAL AUTHOR(S)   Dirk Ruiz and Allen Newell

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept15 TO 91Sept14	89 June	8

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Tower of Hanoi, Soar, unified theories of cognition, strategy change, problem solving

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

SEE REVERSE SIDE

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Dr. Alan L. Meyrowitz	(202) 696-4302	N00014

Abstract

People who solve the Tower of Hanoi start out with a guided trial-and-error strategy and later acquire a recursive strategy, the generally most effective strategy. Protocol data shows that noticing and using subtowers in problem-solving differentiates two subjects who acquired the recursive strategy from one who did not. A working Soar model explains Tower of Hanoi strategy-acquisition by first assuming the basic ability to notice and use subtowers, and then charting the process by which this new knowledge is integrated with existing knowledge to produce the recursive strategy. Of particular importance in the integration is learning to see *nested* subtowers and using simple spatial-manipulation reasoning to figure out how to move those subtowers. The model shows a good qualitative fit to the data, providing support for Soar as a unified theory of human cognition.

Tower-Noticing Triggers Strategy-Change in the Tower of Hanoi: A Soar Model

Dirk Ruiz and Allen Newell

Departments of Psychology and Computer Science
Carnegie Mellon University

People who solve the Tower of Hanoi start out with a guided trial-and-error strategy and later acquire a recursive strategy, the generally most effective strategy. Protocol data shows that noticing and using sub-towers in problem-solving differentiates two subjects who acquired the recursive strategy from one who did not. A working Soar model explains Tower of Hanoi strategy-acquisition by first assuming the basic ability to notice and use subtowers, and then charting the process by which this new knowledge is integrated with existing knowledge to produce the recursive strategy. Of particular importance in the integration is learning to see *nested* subtowers and using simple spatial-manipulation reasoning to figure out how to move those subtowers. The model shows a good qualitative fit to the data, providing support for Soar as a unified theory of human cognition.

THE PHENOMENON

How are new problem-solving strategies acquired? In particular, what new information triggers the acquisition process, and how is that information integrated into problem-solving to produce a new strategy? We address this question within the Tower of Hanoi domain. The puzzle (Figure 1) consists of five disks of graded sizes which in the initial state are sitting on one peg (the Source) to form a tower. The object is to move the disks to the Destination peg while moving only one disk at a time from peg to peg and never placing a larger disk on a smaller. Much is already known about this problem. In particular, it has been found that the Tower of Hanoi can be solved with a small number of well-defined, easily detectable strategies and that people tend to learn new strategies while solving it (Simon, 1975; Anzai & Simon, 1979). These characteristics make the problem ideal for studying strategy-acquisition.

Subjects start out solving the Tower of Hanoi using a guided trial-and-error (GTE) strategy and frequently end up using a recursive strategy (Egan & Greeno, 1974; Simon, 1975; Anzai & Simon, 1979; Ruiz, 1988; VanLehn, 1989). The GTE strategy consists of never moving the same disk twice in a row and never returning a disk to the peg from which it most recently came. The recursive strategy consists of setting a goal to move the largest disk that is not yet on the Destination to the Destination, followed by recursively setting subgoals to move blocking disks out of the way. Prior speculations and models of this change have typically relied on three assumptions (Egan & Greeno, 1974; Simon, 1975; Anzai & Simon, 1979). First, search takes place only in the Tower of Hanoi problem-space. Second, prior knowledge of (domain-independent) means-ends analyses (MEA) is necessary either to guide the acquisition of new
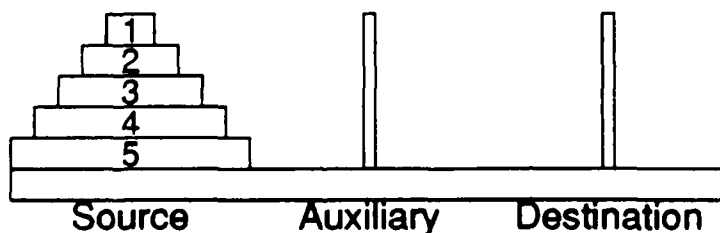

Figure 1: The Five-Disk Tower of Hanoi.

information that will later lead to the development of the recursive strategy, or to directly provide a template for building the recursive strategy. Third, little prior knowledge, drawn from other domains, is assumed. (VanLehn's model sets aside the MEA assumption, but retains the assumptions of a single problem-space and little prior knowledge.) Our theory, a working Soar model (Newell, in press), provides a new explanation of Tower of Hanoi strategy acquisition without having to make these sometimes overly restrictive assumptions.

HUMAN DATA

Our model is based on thinking-aloud protocols taken from three subjects solving the five-disk Tower of Hanoi problem: AS, RN, and PD. AS is Anzai and Simon's well-known (1979) subject, who did the following trials using a physical problem: part of a five-disk, a complete five-disk, a one-disk, a two-disk, a three-disk, a four-disk (these last four problems were experiments initiated by AS), and then two more five-disks. RN and PD were run in our laboratory on an IBM PC; they used special keys to pick up and drop disks. Numbers were printed on the disks to aid identification. They each did five complete trials of the five-disk Tower of Hanoi. Their moves and times were recorded. The following regularities were observed in the protocols:

1. All three subjects initially used the GTE strategy; RN and AS later acquired the recursive strategy. AS acquired it upon starting her three-disk experiment; RN acquired it during his first trial at state _,1234,5 (the Source is blank, Auxiliary has disks 1-4, and 5 is on Destination). While PD was able to solve the Tower of Hanoi, he did not acquire the recursive strategy; he did use lookahead, but that only infrequently (six times).

2. Solving one-disk and two-disk subtowers was trivial for all subjects.

3. Before acquiring the recursive strategy, RN set five subgoals and AS set six. Two of RN's subgoals and three of AS's subgoals were to move a two-disk subtower. In both, the two-disk tower was then planned out and moved. The remaining goals were to move larger towers; they were abandoned in favor of a move generated according to the GTE strategy.

4. Upon or after placing disk 4 on the Destination, all three subjects realized they had made an error and tried to rectify it.

5. Subtowers figured prominently in RN's and AS's pre-recursive problem-solving, but not in PD's pre-lookahead problem-solving. (A subtower is defined as a stack of k consecutive-sized disks, with disk 1 as the first disk.) RN mentioned them explicitly in six statements such as "Now I need to move disks one, two, and three [to the] auxiliary." AS did her aforementioned four experiments solving subtowers. PD did mention subtowers five times in his pre-lookahead problem-solving. These mentionings were either vague or were comments on a tower just about to be completed; they never occurred while planning a move.

6. Recursion was displayed suddenly by both AS and RN, within one move.

7. Recursive reasoning occurred when AS and RN were confronted with subtowers (e.g., the state 5,4,123); there was little mention of goals while moving between subtowers.

SOAR

The theory we shall present is a subtheory within Soar. Soar provides the general theoretical constructs (learning, problem-spaces, memory structure, etc) that support and realize our strategy-change explanation. To understand the subtheory, it is necessary to first understand Soar.

Soar is a general cognitive architecture which models the human cognitive architecture (Newell, in press). Soar has an associative, recognition-based long-term memory (LTM), realized by productions, and a working memory to hold intermediate results of computation. As with humans, the *problem-space* (the problem-states and operators that can be used to solve the problem) is the basis of Soar's cognition (Newell, 1980). Given a goal to solve a problem, Soar makes progress by first deciding on a problem-space for that goal, then deciding on a problem-state, and then deciding on an *operator* to apply to that state. The operator is used to create a new state, to which new operators are applied, and so on. Each decision (problem-space, state, and operator) takes place within a *decision-cycle*, during which candidates are first generated via production match from LTM, followed by the selection of the best candidate. The selection is done by collecting desirability information about each candidate from LTM, and then applying a fixed decision procedure to this information. If Soar has conflicting or insufficient information to make a decision, it sets up a subgoal to resolve the *impasse*. By searching in one or more problem-spaces, Soar generates the needed information and resolves the impasse. Impasses can occur within impasses, leading to a subgoal hierarchy. Soar learns from its experience in resolving impasses by determining which *working-memory elements were responsible for generating the results that resolved the* most recent impasse in the stack. The responsible elements and the generated results become the condition and action, respectively, of a new production, a *chunk*. The chunk will fire when the impasse situation (or any other sufficiently similar situation) occurs in the future. Thus the chunk both avoids impasses and causes transfer of learning. Soar has no pre-built procedures for handling faulty chunks: they must be detected and overriden by deliberate problem-solving.

THE MODEL

The model consists of productions that provide the knowledge Soar needs to use the various problem-spaces. Each problem-space corresponds to a particular cognitive function (e.g., how to generate operators, and how to reason about generated operators). Our description of the model will first be problem-space oriented, with the exception of subtower-noticing, which does not occur in its own problem-space (Figure 2). Then, we will describe the mechanics of the individual models for RN and AS, and finally their general behavior. (PD is not modeled here, but is used simply to supply contrasting data.)

TOH

Tower Of Hanoi moves disks in the real world. TOH is assumed to arise from comprehending the
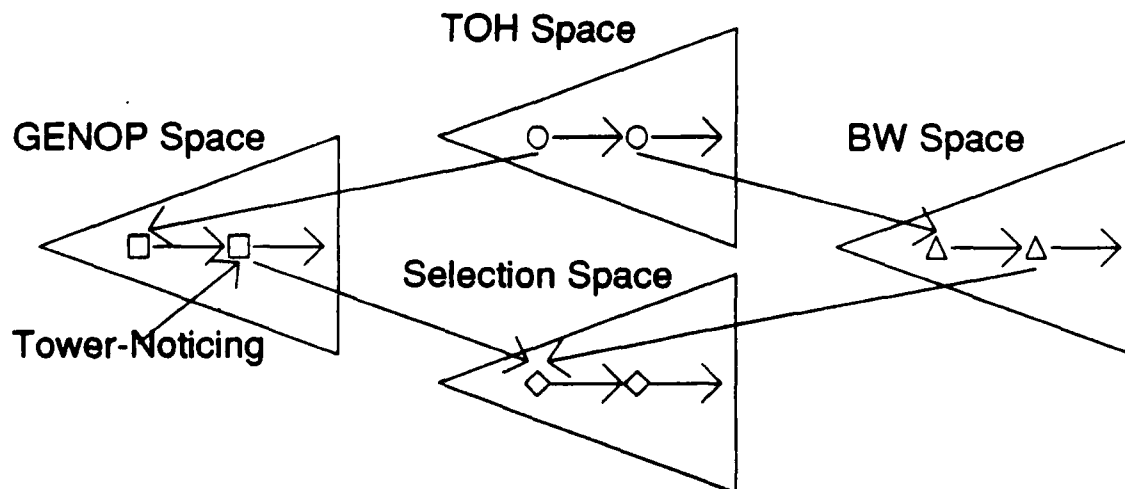

Figure 2: Summary of Problem-Spaces.

problem instructions, which specify how disks may be moved. (Future versions of this model will actually comprehend problem instructions, using the language comprehension system developed by Lewis, Newell & Polk (1989).) TOH translates the physically realizable results of operator generation and reasoning into disk movements. TOH is implemented in Soar with the move-disk operator, which transfers a disk from one peg to another.

GENOP

GENerate OPerator generates move-disk operators for TOH by first choosing a disk to move and then choosing a peg to which to move it. Operator generation is given a separate problem-space because it is a complex cognitive activity, as indicated by the fact that subjects employ several heuristics in generating operators. GENOP uses two heuristics to select disks for movement: do not repetitively move an object, and prefer the largest (non-repetitive) object which can be moved. GENOP employs three heuristics to select pegs: do not block the goal on the first move, do not move an object to the location from which it was last moved, and prefer to move an object to its final location (the Destination peg). Prior to the problem-solver perceiving a stack of consecutive-sized disks as both a unit (a subtower) and as movable, GENOP generates move-disk operators from the tops of the stacks, as they are the only movable disks. This effectively implements the GTE strategy. After a problem-solver has started seeing stacks of disks as movable subtowers, GENOP will generate operators to move the bottom disk of an encountered subtower. This effectively generates the operators needed to begin the recursive strategy. In the Soar model, GENOP has two operators: choose-disk and choose-to-peg. The results of these operators are recorded on the problem-state and then used to form a move-disk operator for TOH.

BW

Blocks World reasons about TOH's unimplementable move-disk operators. BW is so named because it reflects people's ability to do simple spatial-manipulation reasoning, of which blocks-world type problems are a paradigmatic case. This reasoning capability is called into use by the Tower of Hanoi's spatial character. Since spatial-manipulation reasoning is at least able to solve a two-disk problem, BW was built with enough knowledge to do that. BW does three things. First, given an unimplementable move, it determines the *blocking* object using two heuristics: prefer the largest movable object, and prefer the object that blocks the unimplementable move's desired location. Second, BW chooses a peg to which to move the blocking object, using the heuristic that it is best to move the blocking object out of the way of the desired move. Third, BW tries out the assembled move-disk operator. If this results in a new state, then the move-disk operator is implementable and is used to resolve the unimplementable operator's impasse. If no new state is produced, an impasse has occurred and BW is used again to resolve this new impasse. The final result of BW's reasoning is returned to TOH and executed. In the Soar model, BW has one operator for each action: a mark-blocking-disk operator, a mark-receiving-peg operator, and an operator called try-operator. The results of the first two operators are recorded on the state. Try-operator tries out the assembled operator as discussed above. When BW finally reasons out a move, it provides TOH with both the move and the operator it should use next, as determined in the reasoning chain.

Selection

Selection is a default, domain-independent problem-space that collects information about alternatives (usually, operators) and uses that information to choose one of them. The selection-space is derived from people's ability to use heuristics and to make simple choices as a result of applying those heuristics. The selection-space is used when a problem-space encounters an impasse in which it has insufficient information to directly choose between several alternatives. The selection-space applies all relevant heuristics for each alternative separately, and then integrates the results of those heuristics into a single choice, thus resolving the original problem-space's impasse. (The selection-space does not generate

heuristics; it only applies them.) The selection-space has one operator: evaluate-object, which evaluates each alternative and records the results of each evaluation on the selection-space's problem-state.

Subtower-Noticing

Subtower-noticing distinguished RN and AS from PD, and thus is postulated to be the crucial variable in discovering the recursive strategy. Subtower-noticing occurs when the subject realizes that a series of consecutive, stacked disks (starting with disk 1) forms a tower, and makes the assumption that this tower might be moved as a unit. The subject's ability to see consecutive disks as a tower is derived from his/her prior knowledge of such things as towers, pyramids and triangles. The impetus to apply this knowledge to the problem comes from the problem name (the TOWER of Hanoi) plus the frequency with which subtowers occur in the problem. For example, in a perfect solution to a 5-disk problem, 12 of the 32 problem-states will have a subtower sitting by itself on a peg. Most important, the subject will, during
· his/her solution, encounter these subtowers in order of size. For example, s/he may see a two-disk tower followed by a three-disk tower, followed by a four-disk tower. This experience should lead the subject to see the subtowers as being nested, i.e., that a three-disk tower really consists of a two-disk tower sitting on top of disk 3. Seeing subtowers as nested is important because it effectively breaks a subtower down into components about which BW can reason. For, a nested subtower of k disks is really just a two-disk subtower: a $(k - 1)$-disk subtower sitting on disk k. (Of course, the $(k - 1)$-disk tower must itself be a two-disk tower for reasoning to proceed.) In Soar, tower-noticing occurs in the GENOP problem-space upon seeing a k-disk subtower sitting by itself on a peg. The noticing (implemented via productions) results in a minor change to the problem-state: the tower's bottom disk is marked as a tower and as movable. A chunk gets built that can then label that tower in any context, including when it is the subtower of another tower. Repeated experience with single subtowers thus gives Soar the ability (i.e., the chunks) to see nested subtowers. Since these chunks allow the use of BW, they (and not the subtower-noticing productions) are directly responsible for the development of the recursive strategy.

Model Mechanics

Two variant models were created, one each for RN and AS. The GENOP, BW, and TOH problem-spaces were the same in the two models. The productions constituting these problem-spaces were integrated into LTM before solving the Tower of Hanoi. The two models differed in three areas. First, RN's model was given the ability to label subtowers as subtowers before it was given the ability to label them as movable, while AS' model was given the two abilities simultaneously. This corresponds to the fact that RN noticed subtowers well before trying to move them, whereas AS noticed and used subtowers at the same time. In both models, the relevant productions were integrated into LTM at the indicated point in the human subjects' problem-solving. Second, AS required several additional productions to model the different initial states in her one-disk through four-disk problems. These were integrated into LTM before the start of each problem. Third, a small set of special-case productions (two for AS and one for RN) modeled instances in which the subjects used reasoning processes outside of the scope of the GENOP, BW, and TOH problem-spaces. In AS' model, one production was used to halt her simulation after placing disk 4 on the Destination in the first trial, mimicking AS' giving up midway on her first trial. Another production mimicked AS' second-trial realization that disk 1 had to go to the Destination on the first move. Finally, one production mimicked RN's violation of the disk non-repetition heuristic at state 5,_,1234. The first of these productions was introduced along with the three problem-spaces; the last two were introduced at the appropriate points in the problem-solving. Effectively, the two models only explained moves that corresponded to a strict use of the recursive and GTE strategies. Learning was always on; the models therefore learned continuously from their experience.

Model Behavior

Upon starting to solve the problem, the models used the TOH problem-space. Since TOH cannot

generate operators, it encountered impasses and used GENOP to supply the needed move-disk operators. At this point, the models did not notice subtowers, and therefore generated implementable operators according to the GTE strategy. Upon learning about subtowers, GENOP began generating operators to move (the bottom disks of) subtowers. Since such operators were not directly implementable, impasses resulted and BW was used to reason out how to make progress towards applying the operators. BW tried to generate an operator to move the blocking disk/tower out of the way. If BW produced an implementable move-disk operator, the new operator was used in TOH. If BW produced an unimplementable operator, another impasse resulted and BW was once again used to reason about the new unimplementable operator. This successive use of BW on a single move produced the observed recursion. BW's first implementable result was used to continue progress in TOH. The selection-space was used every time GENOP or BW had to choose between several versions of an operator. The selection-space applied the heuristics discussed above to make the choices. Learning had three major effects on the model's behavior. First, it produced the chunks that noticed nested subtowers. Second, it abbreviated, with time, the amount of processing needed to use both the GTE and recursive strategies. Third, it eventually built chunks that directly generated move-disk operators in TOH, thus bypassing strategic processing.

MODEL AND DATA: THE FIT

Problem-Solving Fit

The models show a good quantitative fit to the subjects' external problem-solving behavior. Moves made by the model corresponded to 77% and 67% of AS's and RN's first-trial moves, respectively. In the remaining trials, the correspondence was almost always 100% with the exception of AS's second trial (94%) and RN's last trial (97%). Unmodeled moves were the result of errors or error-recovery on the subjects' part, both of which deviated from a perfect GTE or recursive sequence; these moves either had no analog in the models, or were mimicked with the special-purpose productions described above. (Both types of moves were counted against the models.) To test the models further, RN's move-times were correlated with the number of decision-cycles that his model required to make the corresponding moves. (Unmodeled moves were not included in this correlation.) The correlations, in order of trials, were $r = 0.69, 0.61, 0.85, 0.45$ (0.75), and 0.01. The Trial 4 correlation was low because RN remembered the first move and executed it directly, whereas the model reasoned it out; without this outlier, the correlation goes to 0.75. The final correlation was nil (and should be), because both RN and the model had low variance. Move-time data was not reported for AS in Anzai & Simon (1979) so no correlations could be calculated.

The models' problem-solving strategies displayed a good qualitative fit to the subjects'. We simulated a pure form of the GTE and recursive strategies. Therefore, the models showed some differences: their GTE showed no occasional goal-setting and their recursive strategy showed excess goal-setting between subtowers. Within these boundaries, the models' GTE and recursive strategies showed the same type of reasoning and behavior as AS and RN. The protocols might be fit more closely by remembering more of BW's reasoning chain or using the BW problem-space earlier. However, a closer fit would not change the basic result, i.e., that the recursive strategy arises from noticing and using (nested) subtowers in problem-solving.

Srategy-Change Fit

The models closely fit the qualitative aspects of the subjects' strategy-change. AS's and RN's models acquired the recursive strategy at the same point that AS and RN did. RN's model, like RN, acquires the recursive strategy at state _,1234,5; AS's model, like AS, acquires it during its 3-disk experiment. In both cases, the model's acquisition was the result of having correctly mimicked subtower-noticing and use. RN's model had been learning to notice subtowers for the same amount of time that

RN did; upon acquiring the ability to mark labeled towers as movable, the model was immediately able to make use of that information. AS's model had had previous experience with a two-disk tower, and thus saw the three-disk tower as a nested tower (a two-disk tower sitting on disk 3), allowing it to apply BW. Thus, the switch to the recursive strategy is not due to the subtower-noticing productions per se, but to the chunks that allow the models to see the subtowers as nested and to Soar's ability to recursively use problem-spaces. Finally, the models, like RN and AS, acquired the recursive strategy suddenly. This is because the models treat movable objects (disks or subtowers) alike: as soon as subtowers are noticed, they can immediately be reasoned about.

Learning Fit

The learning displayed by the models showed a good qualitative fit to the subjects'. Besides noticing subtowers, the models' learning did two major things: it abbreviated strategic processing and it eventually caused well-learned moves to be executed directly, without the need for any strategic processing. The abbreviation of strategic processing appeared in the protocols as decreased verbalization over trials, as well as corresponding decreases of move-times in RN's protocol. Making well-learned moves directly executable brought the model more in line with the sparse recursive reasoning displayed by the subjects. After the moves between major subtowers had been well-learned, the model, like the subjects, only reasoned out moves when confronted with a subtower.

CONCLUSIONS

Our model has provided a simple answer to the question of Tower of Hanoi strategy-acquisition. It comes about because people notice nested subtowers and use spatial-manipulation reasoning to move them. This latter capability is cast in terms of physical objects in general, and so is able to work with either disks or towers once these have been noted as relevant to the task at hand. In positing this answer, our model has bypassed the need for some of the basic assumptions of previous models and speculations. Our model, like previous models, works by problem-space search. However, our model works in multiple problem-spaces, not one, and thus claims that people (who can use multiple types of reasoning on a single task) do likewise. The claim of multiple problem-spaces is both a specific claim of our model and a general claim of Soar, which processes information in many problem-spaces.

Second, we have reduced the types of information that prior models claimed had to be learned. Like previous models, ours learns continuously from its experience. But, the crucial information that allows the switch to the recursive strategy is noticing (nested) subtowers. This paper has described how this knowledge is incorporated into problem-solving to produce a new strategy; future work will tackle the mechanics of subtower-noticing per se.

Third, our model has eliminated prior knowledge of domain-independent MEA as the cause of strategy-change. While our model certainly behaves according to MEA in the BW problem-space, that MEA is a direct result of domain-specific knowledge, and is not necessarily transferable to non-spatial-manipulation domains. The recursion characteristic of Tower of Hanoi solutions comes about because of Soar's ability to recursively use problem-spaces to resolve impasses. We have therefore set up a strong alternative hypothesis to strategy-adaptation: strategy-building. This claim about MEA is a general Soar claim, as Soar has nothing corresponding to a domain-independent MEA. Rather, Soar's use of weak methods stems from its task knowledge against the background of its use of multiple problem-spaces (Laird & Newell, 1983)

Fourth, our model directly relies on prior knowledge of a specific domain: spatial-manipulation problems. We have therefore taken this task out of the realm of knowledge-lean tasks, and made it more knowledge-intensive, where the knowledge used is knowledge of spatial relationships and operators. In

so doing, we have blurred the boundaries of the traditional toy-task category.

Finally, the work done here will generalize to other task domains. The fundamental insight, that strategy-change comes about via noticing aggregate problem features and attempting to operate on them. is certainly empirically verifiable, and therefore amenable to modeling, in other tasks. The GENOP problem-space might be easily extended to other tasks, thus providing a source of models and ideas about people's default strategies. Finally, the BW problem-space might be expanded to include many other spatial-manipulation tasks, and thus might be the starting point for a single Soar theory of puzzle problem-solving.

To conclude, this model derives its explanatory power from being a subtheory of Soar. Soar provides the ability to search in problem-spaces, the learning, and the theory of how knowledge is transmitted between problem-spaces. Our task as theorists has been to carefully specify the knowledge people have about the Tower of Hanoi. What Soar has thus done is not only ease our burden as theorists, but reduce our theoretical degrees of freedom as well. In return, the success of this model supports Soar as a unified theory of human cognition.

ACKNOWLEDGEMENTS

REFERENCES

Anzai, Y., & Simon, H. A. (1979). The theory of learning by doing. *Psychological Review, 86(2)*, 124-140.

Egan, D. E., & Greeno, J. G. (1974). Theory of rule induction: Knowledge acquired in concept learning, serial pattern learning, and problem solving. In L. W. Gregg (Ed.), *Knowledge and cognition*, Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Laird, J. & Newell, A. (1983) A universal weak method: Summary of results. In *Proceedings of IJCAI-83*, Los Altos, CA: Kaufman.

Lewis, R., Newell, A. & Polk, T. (1989). *Towards a Soar theory of taking instructions for immediate reasoning tasks*. Manuscript submitted for publication.

Newell, A. (1980): Reasoning, problem solving, and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.), *Attention and Performance VIII*, Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Newell, A. (in press). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Ruiz, D. (1988). Learning and problem-solving: What is learned while solving the Towers of Hanoi (Doctoral dissertation, Stanford University, 1987). *Dissertation Abstracts International, 42*, 3438B.

Simon, H. A. (1975). The functional equivalence of problem-solving skills. *Cognitive Psychology, 7*, 268-288.

VanLehn, K. (1989). *Learning events in the discovery of problem solving strategies*. Manuscript submitted for publication.