A Rational Analysis of Alternating Search and Reflection Strategies in Problem Solving

Niels Taatgen (`n.a.taatgen@bcn.rug.nl`)
Cognitive Science and Engineering, University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen, the Netherlands

Abstract

In this paper two approaches to problem solving, search and reflection, are discussed, and combined in two models, both based on rational analysis (Anderson, 1990). The first model is a dynamic growth model, which shows that alternating search and reflection is a rational strategy. The second model is a model in ACT-R, which can discover and revise strategies to solve simple problems. Both models exhibit the explore-insight pattern normally attributed to insight problem solving.

Search vs. Insight

The traditional approach of problem solving is one of *problem space search* (see for example Newell & Simon, 1972). Solving a problem means no more or less than finding an appropriate sequence of operators, that transform a certain initial state into a state that satisfies some goal criterium. Problem solving is difficult if the sequence needed is long, if there are many possible operators, or if there is no or little knowledge on how to choose the right operator.

A different approach to problem solving is that the crucial process is *insight* instead of search. This view also has a rich tradition, rooted in Gestalt psychology. According to the insight theory, the interesting moment in problem solving is when the subject suddenly "sees" the solution, in a moment when an "unconscious leap in thinking" takes place. Instead of gradually approaching the goal, the solution is reached in a single step, and reasoning efforts before this step have no clear relation to it. In this sense problem solving is often divided into four stages: exploration, impasse, insight and execution.

Insight can be viewed in two ways: as a special process, or as a result of ordinary perception, recognition and learning processes (Davidson, 1995). Despite the intuitive appeal of a special process, the latter view is more consistent with the modern information-processing paradigm of cognitive psychology, and much more open to both empirical study and computational modeling.

Both the search and the insight theory select the problems to be studied in accordance with their own view. Typical "search"-problems involve finding long strings of clearly defined operators, as in the eight puzzle, the towers-of-hanoi task and other puzzles, often adapted from artificial intelligence toy domains. "Insight"-problems on the other hand, can be solved in only a few steps, often only one. Possible operations are often defined unclearly, or misleadingly, or are not defined at all. A typical insight problem is the nine-dots problem, in which nine dots in a 3x3 array must all be connected using four connected lines. The crucial insight is the fact that the lines may be extended outside the 3x3 boundary. Other insight problems are the box-candle problem and several types of matchstick problems (see for example Mayer, 1995). Due to this choice of problems, both evidence from insight and search experiments tend to support their respective theories. Both theories ignore aspects of problem solving. The search theory seems to assume that subjects create clear-cut operators based on instructions alone, and that subjects do not reflect on their own problem-solving behavior, while the insight theory assumes all processing that happens before the "insight" occurs has hardly any relevance at all. So probably search and insight are both aspects of problem solving, and the real task is to find a theory of problem solving that combines the two (Ohlsson, 1984). One such view sees insight as a representational change. Search is needed to explore the current representation of the problem, and insight is needed if the current representation appears not to be sufficient to solve the problem. In this view, search and insight correspond to what Norman (1993) calls *experiential* and *reflective* cognition. If someone is in experiential mode, behavior is largely determined by the task at hand and the task-specific knowledge the person already has. In reflective mode on the other hand, comparisons between problems are made, possibly relevant knowledge is retrieved from memory, and new hypotheses are created. If reflection is successful, new task-specific knowledge is gained, which may be more general and on a higher level than the existing knowledge.

The scheduling task

An example of a task in which both search and insight are necessary is scheduling. Figure 1 shows an example of a scheduling task used in our experiments. The goal is to assign a number of tasks (6 in the example) to a number of workers (2 in the example), satisfying a number of order constraints. A solution to the example in figure 1 is to assign

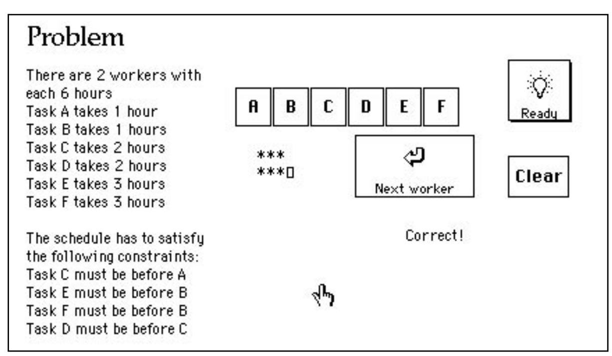

Figure 1: Example of a scheduling experiment

DEA to the first worker, and FCB to the second. An experiential strategy for this problem is to take one of the constraints, for example C must be before A, and assign them to a worker directly following each other, so for example assign CA to the first worker. A reflective strategy involves some inference. For example, from the facts that both E and F have to be done before B, that E and F each take 3 hours, and that each worker has only 6 hours, it can be deduced that E and F cannot be assigned to the same worker. Furthermore, since each worker has 6 hours, either each worker gets a task of 3, 2 and 1 hours, or one worker gets both 3 hour tasks and the other the rest. Since the latter option has already been ruled out by the fact that E and F can't be assigned to the same worker, the former has to be correct. When the subject has made these inferences a few times for different problems, they can become part of the experiential strategy.

Protocol analysis of subjects solving these problems show that all subjects start with an experiential strategy, and only later on switch to a reflective strategy. So in a sense this reflects the explore-impasse-insight-execute pattern described in the literature about insight. Some, but not all of the subjects show some sort of impasse, during which they stop searching, just stare at the screen for a minute, and then try a new approach. Furthermore, there is no difference between the explore and the execute stage: the subject just searches on, using the knowledge gained by reflection. Sometimes further reflection is needed to reach a solution.

In this paper two models are presented that explore the distinction between search and reflection. Both models are based on Anderson's theory of rational analysis (Anderson, 1990). According to rational analysis, subjects choose strategies based on a cost-benefit analysis: the strategy that has the lowest expected cost and the highest chance of success is selected in favor of others. The first model is a dynamic growth model, in which the trade-off between search and reflection is modeled in a course-grained way. Dynamic models are used in developmental psychology to describe developmental paths, for instance a model that describes stage-wise increases in knowledge (Van Geert, 1994). The second model is an ACT-R model, in which the competition between individual strategies is modeled on a number of concrete tasks.

A dynamic growth model

Why would subjects initially prefer the experiential strategy in the scheduling problem? The reflective strategy seems to be much more powerful. There are several reasons for this. A first reason is that reflective reasoning takes more effort. To be successful, several aspects of the task must be combined and kept in memory. Additional knowledge must be retrieved from memory and it may be necessary to seek analogies with other problems. A second reason is that it is not immediately evident to the subject that the experiential strategy will be unsuccessful. The problems in the experiment were chosen so that the experiential strategy alone wouldn't work, but subjects didn't know this. So why not try the strategy which takes the least effort first? A third reason is, that as a subject starts with a new type of problem, he has only read instructions and has maybe seen an example problem. So he first has to learn the basic rules and operators by experience, before he can attempt any higher level strategies.

The model

According to rational analysis strategies are chosen with respect to their expected outcome, according to the following equation:

$$\text{Expected outcome of strategy } s = P_s G - C_s$$

In this equation, P is the estimated chance of reaching the goal using this strategy, G is the expected value of the goal, and C is the estimated cost of reaching the goal using this strategy.

The model will attempt to predict how search and reflection will alternate while solving a problem. This model is quite course-grained in the sense that the knowledge of the system with respect to a certain task is summarized in two variables L_1 and L_2. L_1 is a measure for the amount of basic task-knowledge in the model, for example in the case of the scheduling task an operator to add a task to an existing plan and knowledge to judge whether a solution is correct. L_2 corresponds to the amount of higher-level knowledge in the system, for example the fact that it is a good idea to look how the tasks add up to the amount of time the workers have available. If a subjects starts with a new problem, we assume that both variables have a small value. They can however increase, because the subject builds up knowledge while problem solving. The assumption of the model will be, that search will increase the amount of basic knowledge, represented by L_1, and reflection will increase the amount of higher-level knowledge, represented by L_2. The following equations show how L_1 and L_2 grow in time, and are based on the equation used by Van Geert (1994):

If the strategy in step *i*-1 is search:

$$L_1(i) = L_1(i-1) + R_1\left(1 - \frac{L_1(i-1)}{L_{1max}}\right)$$

else L_1 keeps its value, so $L_1(i) = L_1(i-1)$. R_1 is a constant that controls the rate of growth, and L_{1max} is the maximum possible value for L_1. The fraction at the end of the equation ensures that L_1 doesn't exceed its maximum value.

The equation for L_2 is slightly more complicated, because the increase in value depends on the current value of L_1: we can only gain higher-level knowledge if we have enough basic knowledge.

If the strategy at step *i*-1 is reflection:

$$L_2(i) = L_2(i-1) + S_{12} \cdot L_1(i-1)\left(1 - \frac{L_2(i-1)}{L_{2max}}\right)$$

else $L_2(i) = L_2(i-1)$. Again, L_{2max} is the maximum possible value for L_2. The constant S_{12} (support) controls the influence of basic knowledge on the increase of higher level knowledge.

Whether the strategy at step *i* will be search or reflection is determined by their respective expected outcomes:

$$\text{Expected outcome of search} = P_{search}(i) \cdot G - C_{search}$$

$$\text{Expected outcome of reflection} = P_{ref} \cdot G - C_{ref}(i)$$

The strategy with the highest expected outcome will of course be chosen. In these equations G, C_{search} and P_{ref} are constants, but $P_{search}(i)$ and $C_{ref}(i)$ will vary in time.

The chance that search will reach the goal is dependent on the amount of knowledge and the current evaluation of this knowledge:

$$P_{search}(i) = \frac{L_1(i)P_1(i) + wL_2(i)P_2(i)}{L_1(i) + wL_2(i)}$$

The w constant determines how much more useful higher-order knowledge is with respect to basic knowledge. $P_1(i)$ is the contribution to the chance of success of level 1 knowledge, and $P_2(i)$ the contribution of level 2 knowledge. The chances of success increase as knowledge increases, but decrease over time if the goal is not reached. Both $P_1(i)$ and $P_2(i)$ can be calculated using:

$$P_j(i) = p_{decay}\frac{L_j(i-1) \cdot P_j(i-1) + L_j(i) - L_j(i-1)}{L_j(i)};(j = 1, 2)$$

p_{decay} represents the decay in chance of success, and has typical values between 0.95 an 0.99 if the strategy in step *i* was search and the goal hasn't been reached. Otherwise $p_{decay} = 1$.

The cost of reflection depends on two factors: the cost is higher if there is less basic knowledge, and the cost is higher if there is already a lot of higher level knowledge:

$$C_{ref}(i) = C_{base} + \left(c_1\frac{L_{1max}}{L_1(i)}\right) + \left(c_2\frac{L_2(i)}{L_{2max}}\right)$$

Finally we have to say something about time, since we have talked about "steps" in the previous discussion. Each step takes an amount of time which can vary. So, following the ACT-R intuition that cost and time are related to each other, we take the estimated cost of the strategy at step *i* as the amount of time step *i* takes:

$$T(i) = T(i-1) + C(i)$$

where $C(i)$ is either C_{search} or $C_{ref}(i)$, dependent on the strategy at step *i*.

Results

If we choose appropriate constants and starting values for the variables described above, we can calculate the increase in knowledge over time. Note that the model assumes that the goal is never reached, so the results simulate a subject that never succeeds in reaching the goal. Figure 2 shows the value of L_1 and L_2 with respect to T. The corresponding evaluations for search and reflection are shown in figure 3. At the start of the task, search is superior to reflection, but as search fails to find the goal, and the basic (level 1) knowledge increases, reflection becomes more and more attractive up to the point (at T=127) where reflection wins from search. Since reflection leads to an increase of level 2 knowledge, search becomes again more attractive (using the newly gained knowledge), and since the cost of reflection increases

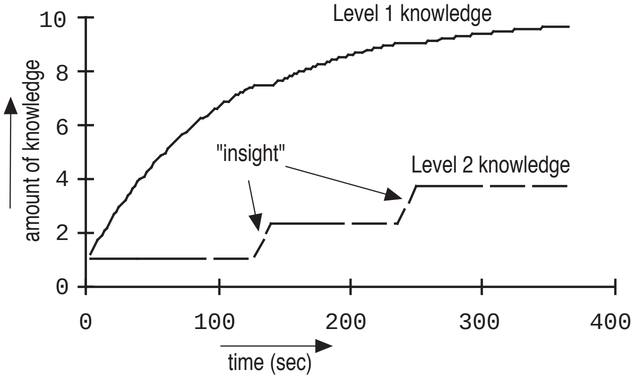

Figure 2: Value of level 1 and level 2 knowledge for G=20

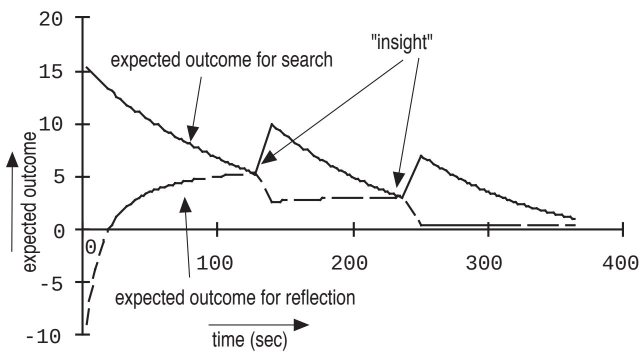

Figure 3: Evaluations of search and reflection for G=20

with the amount of level 2 knowledge already present, reflection becomes less attractive. As a result search will again dominate for a while, up to T=385 where reflection wins again. We assume problem solving continues until both expected outcomes drop below zero, since then neither strategy has a positive expected outcome. In the example this is the case at T=510.

Figure 2 and 3 show the results of the model for G=20. As noted, G is the value of the goal. So using a lower value for G corresponds to the fact that a subject values reaching the goal less, or the fact that a subject is less motivated. If we calculate the model for G=15 we get the results as depicted in figure 4 and 5. The result is that reflection occurs only once, and later (at T=203). Furthermore, at T=363 both evaluations drop below zero, so a less motivated individual gives up earlier. If G is further decreased to 12, no reflection at all takes place, and the give-up point is at T=237.

Discussion

The dynamic growth model nicely describes the phenomena around insight in the literature and in our experiments. Furthermore, it explains why this behavior is rational. It also predicts changes in strategy due to motivational factors. It however poses new questions. What is the nature of the basic and higher-level knowledge? How will the model behave if the goal is reached at some point? What mechanism is responsible for gaining new knowledge, and how is it represented? The second model we will discuss in this paper will

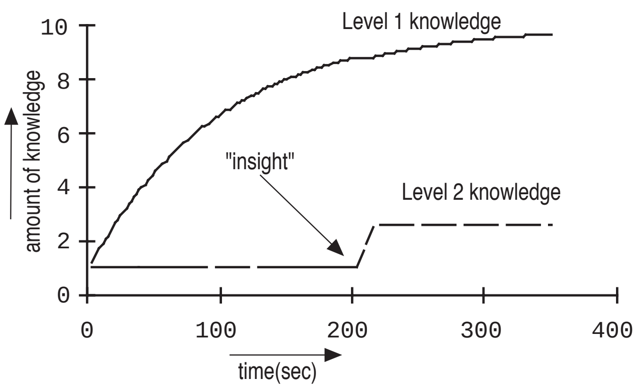

Figure 4: Value of level 1 and level 2 knowledge for G=15

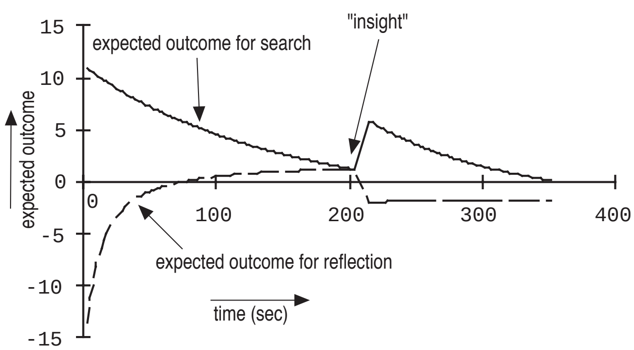

Figure 5: Evaluations of search and reflection for G=15

address some of these questions. This model can be seen as a more detailed version of the dynamic growth model.

An ACT-R model of learning and revising task-specific knowledge

ACT-R is an architecture of cognition developed by Anderson and his colleagues (Anderson, 1993; Lebière, 1996), based on the theory of rational analysis. ACT-R has two long term memory stores, a declarative memory, where knowledge is represented using a frame-like representation, and a procedural memory, where knowledge is represented by production rules. One of the ingredients that ACT-R uses for conflict resolution is the expected outcome of a production rule, in the same manner as described in the previous section. So if several production rules can fire, the rule with the highest $PG - C$ will generally win the competition. Along with the rule the history of successes and failures and the past costs of a rule are maintained to be able to calculate its expected outcome.

In the ACT-R architecture, new production rules can be learned by the analogy mechanism. It involves generalization of examples in declarative memory whenever a goal turns up that resembles the example. The examples are stored in specialized elements in declarative memory, dependency chunks, that contain all the information needed: an example goal, an example solution, facts (called constraints) that need to be retrieved from declarative memory to create the solution, and sometimes additional subgoals that must be satisfied before the solution applies.

Although the ACT-R theory specifies how new production rules are generated from examples, it does not specify where the examples come from. But since examples are just elements in declarative memory, they can be created by production rules. If we give a subject a new task, he will generally have no task-specific rules for the task, but will have to rely on general rules to acquire them. So the schema to produce task-specific production rules will be as in figure 6.

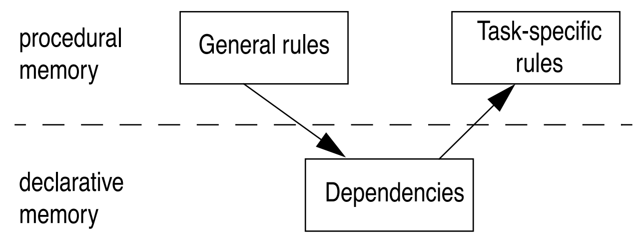

Figure 6: How to learn new production rules in ACT-R

The general rules themselves need of course information to work with. Several sources of information may be available, which must be present in declarative memory, since production rules cannot directly inspect other production rules. Possible sources of information are:

- Task instructions and examples
- Relevant facts and biases in declarative memory
- Feedback
- Old goals and dependencies for the same problem

As both general and task-specific rules are in a constant competition with each other, they play the same role as the search and reflection strategies in the dynamic growth model. If ACT-R uses task-specific rules, this corresponds to a search-like strategy, and when it uses general rules, this corresponds to reflection. So there is no real difference in ACT-R performance between search and reflection, except that general rules will often retrieve more low-activated elements from declarative memory, which makes them slow and expensive. Since using general rules has a higher cost, task-specific rules will win the competition if they prove to lead to success.

The model

In Taatgen (1996) an example of a model that learns its own task-specific rules is described. It uses two sets of general rules, one that creates an example of retrieving a certain property of the task, and one that creates an example of combining the task with a fact in an attempt to predict the answer. The example task is a beam with both weights and labels on each arm (figure 7). Only the weights have any relevance to the outcome. The strategies that do the task are depicted in figure 8 and figure 9. The property-retrieval strategy creates an example of retrieving one of the available properties, in the case of the beam weight or label. The example will be compiled into a production rule by ACT-R's analogy mechanism. If the new rule doesn't lead to successful predictions, which is the case when label is selected, its evaluation will drop until it loses the competition with the

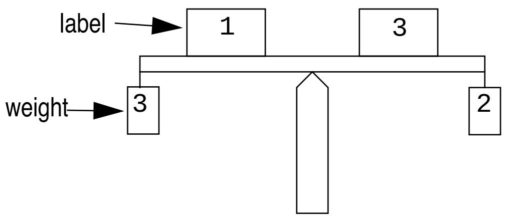

Figure 7: The beam task.

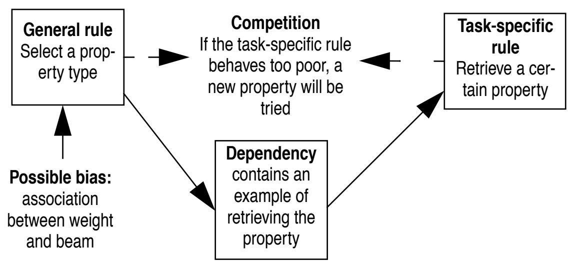

Figure 8: The property-retrieval strategy

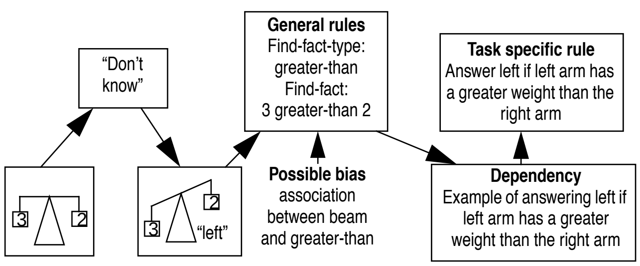

Figure 9: The find-fact-on-feedback strategy

general rule that wants to create a new example. The second strategy, find-fact-on-feedback, is demonstrated in figure 9. If the model has decided it will retrieve the weights, it still cannot predict an answer, because it doesn't even know what the available answers are. So a "I don't know" production fires, after which the environment hopefully will give some feedback. Suppose we have a child facing a real beam, it can see that the answer is "left". The strategy then tries to find some fact in declarative memory that can help to predict the answer. This can be an arbitrary fact, but since "beam", "2", "3" and "left" are all part of the goal, ACT-R ensures that facts containing these elements, or having associations with them, are likely candidates. So 3 is-greater-than 2 is a possible candidate, particularly if there is already an association between beam and greater-than, i.e. the child already knows that beams have something to do with the fact that one thing is greater than another.

Results

Simulations of the model, discussed in detail in Taatgen (1996), show that it can indeed infer the correct rules for the beam task. If the model already has an association between beams and weight and between beams and greater-than, the correct rules can be inferred using only a few examples. If the model has no prior associations at all, it may need as much as 40 examples, and in some runs it cannot even find the correct rules at all.

When the model starts out with a wrong hypothesis, for example that the labels have to be used to predict the outcome, it shows a behavior similar to the explore-impasse-insight-execute scheme: it first learns a lot of irrelevant rules to predict the answer using the labels, then reaches a stage in which it tries to explain a single example over and over again but fails in doing this, after which it rejects the rule that examines the labels, creates a rule that examines the weights, and quickly derives the rest of the rules needed within a few trials. Figure 10 shows an estimation of the time spent at

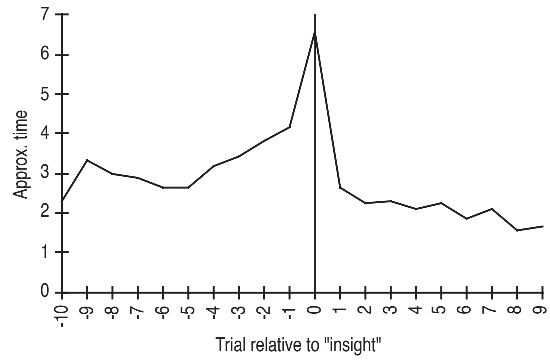

Figure 10: Estimated time spent at each trial before and after the "insight" trial

each trial before and after the moment the model creates the rule to examine weight instead of label.

Same model, other task: discrimination-shift

Another interesting property of the model is that its rules are general, and can be applied to other tasks. A task that can be modeled using the same production rules is discrimination-shift learning (Kendler & Kendler, 1959). Figure 11 shows an example of this task: subjects have to learn to discriminate the four stimuli in two reinforcement categories, for example white is positive and black is negative. After the subjects has made 10 consecutive correct predictions, the reinforcement scheme is changed: either a reversal-shift, in which all stimuli that received previous positive reinforce-

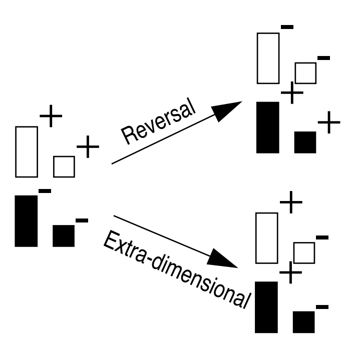

Figure 11: Example of discrimination-shift learning

ment get negative reinforcement and vice-versa, or an extra-dimensional shift, in which the dimension is changed on which the reinforcement is given, in the example from white to large. It turns out that adults and older children are faster at learning the reversal-shift condition, while young children and animals are faster at the extra-dimensional shift. Figure 12 shows the results of an experiment by Kendler and Kendler (1959). The ACT-R model of adult behavior uses the

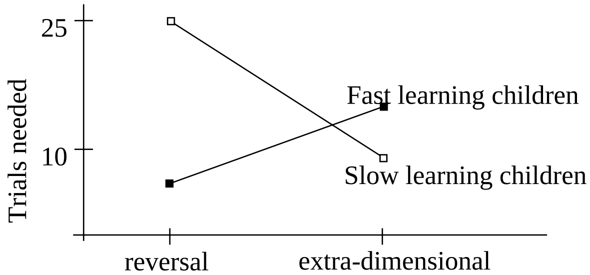

Figure 12: Trials needed to learn the discrimination-shift

same 8 production rules as used in the beam-task, implementing the property-retrieval and find-fact-on-feedback strategies. The small-child/animal model uses only 2 of the 8 production rules, implementing a limited find-fact-on-feedback strategy. The results of these models are shown in figure 13. Although the models do not mimic the subjects results precisely, the general effects are in the same direction.

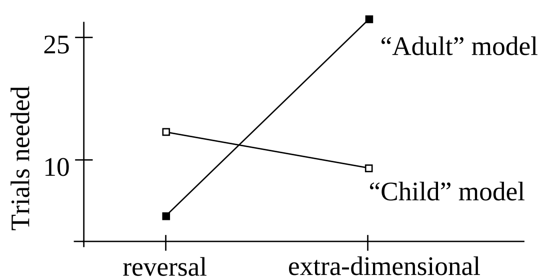

Figure 13: Results of the ACT-R model on the discrimination-shift task

Despite the fact that the discrimination-shift task is generally not considered to be an insight problem, it nevertheless requires the subject to notice that something has changed, and to discover the new relations. So it can be seen, in a sense, as an elementary insight problem.

Discussion

The ACT-R model addresses some of the questions posed by the dynamic growth model. ACT-R itself already answers some of the questions: how knowledge is represented, and what the effects of success or failure are. The question how new knowledge can be acquired isn't fully answered by the ACT-R theory. The model presented here is an attempt to supply the first steps to an answer, since the strategies implemented in the model can learn several different tasks. The discrimination-shift model is interesting in the sense that it shows that an adult model can be changed into a child model only by deleting some productions rules. This shows that the general rules themselves aren't hard-wired, but must be learned as well, perhaps using the same mechanisms as needed for task-specific knowledge. So a next step will be to find out what other general rules people employ, and how general rules themselves can be learned.

Conclusions

So, what should we do when we have to solve a new problem? Just search for a solution or just think hard and hope for an insight? According the dynamic growth model, the most rational thing to do is a proper alternation of the two. The model allows manipulation of several parameters, like the value of the goal as has been discussed. But other parameters can be changed as well, for instance the amount of prior knowledge, importance of level 2 knowledge, etc., allowing for new predictions.

The ACT-R model shows how aspects of the dynamic growth model can be implemented using real knowledge representations instead of variables. Although both knowledge needed for reflection and for search are represented by production rules, there are differences between the two: reflection knowledge is general, and is relatively costly to use, and search knowledge is task-specific and cheap, but possibly insufficient to reach the goal.

References

Anderson, J.R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum.

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

Davidson, J.E. (1995). The suddenness of insight. In R.J. Sternberg & J.E. Davidson (Eds.), *The nature of insight*. Cambridge, MA: MIT Press.

Kendler, T.S. & Kendler, H.H. (1959). Reversal and nonreversal shifts in kindergarten children. *Journal of experimental psychology, 58*, 56-50.

Lebière, C. (1996). *ACR-R 3.0: A users manual.* Pittsburgh, MA: Carnegie Mellon University.

Mayer, R.E. (1995). The search for insight. In R.J. Sternberg & J.E. Davidson (Eds.), *The nature of insight*. Cambridge, MA: MIT Press.

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

Norman, D.A. (1993). *Things that make us smart*. Reading, MA: Addison-Wesley.

Ohlsson, S. (1984). Restructuring revisited II. An information processing theory of restructuring and insight. *Scandinavian Journal of Psychology, 25*, 117-129.

Taatgen N.A. (1996). Learning and revising task-specific rules in ACT-R. In U. Schmid, J. Krems & F. Wysotzki (Eds.), *Proceedings of the First European Workshop on Cognitive Modeling* (report no. 96-39, pp. 31-38). Berlin, Technische Universitaet Berlin, Fachbereich Informatik.

Van Geert, P. (1994). *Dynamic systems of development: Change between complexity and chaos.* London: Harvester Wheatsheaf.