The ACT-R Theory and Visual Attention

John R. Anderson
Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213
ja@cmu.edu

Michael Matessa
Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213
matessa@cmu.edu

Scott Douglass
Carnegie Mellon University
Department of Psychology
Pittsburgh, PA 15213
sd3n@andrew.cmu.edu

Abstract

The ACT-R production-system theory (Anderson, 1993) has been extended to include a theory of visual attention and pattern recognition. Production rules can direct attention to primitive visual features in the visual array. When attention is focused on a region, features in that region can be synthesized into declarative chunks. Assuming a time to switch attention of about 200 msec, this model proves capable of simulating the results from a number of the basic studies of visual attention. We have extended this model to complex problem-solving like equation solving where we have shown that an important component of learning is acquiring more efficient strategies for scanning the problem.

Theories of higher-level cognition typically ignore lower-level processes such as visual attention. They simply assume that lower-level processes deliver some relatively high-level description of the stimulus situation upon which the higher-level processes operate. This certainly is an accurate characterization of our past work on the ACT-R theory (e.g., Anderson, 1993). The typical task that ACT-R has been applied to is one in which the subject must process some visual array—the array may contain a sentence to be recognized, a puzzle to be solved, or a computer program being written. We have always assumed that some processed representation of this visual array is placed into working memory in some highly encoded form and we modeled processing given that representation.

The strategy of focusing on higher-level processes might seem eminently reasonable for a theory of higher-level cognition. However, the strategy creates two stresses for the plausibility of the resulting models. One stress is that by assuming a processed representation of the input the theorists are granting themselves unanalyzed degrees of freedom in terms of choice of representation. It is not always clear whether the success of the model depends on the theory of the higher-level processes or the choice of the processed representation. The other stress is that the theorist may be ignoring significant problems in access to that information which may be contributing to dependent variables such as accuracy and latency. For instance, the visual input may contain more information than can be held in a single attentional fixation, and shifts of attention (with or without accompanying eye movements) may become a significant but ignored part of the processing. For these reasons we have been encouraged to join the growing number of efforts (e.g., Kieras & Meyer, 1994; Wiesmeyer, 1992) to embed a theory of visual processing within a higher-level theory of cognition. The choice to focus on vision is largely strategic—reflecting the fact that most of the tasks that ACT-R has modeled involved input from the visual modality. To be more exact, most tasks have involved processing input from a computer screen and so we have developed a theory of the processing of a computer screen.

It is important to define our approach to the problem from the outset: We require a theory of visual attention and perception which is psychologically plausible but it is not our intention to propose a new theory of visual attention and perception. Therefore, we have embedded within ACT-R a theory which might be seen as a synthesis of the spotlight metaphor of Posner (1980), the feature-synthesis model of Treisman (Treisman & Sato, 1990), and the attentional model of Wolfe (1994). What this model does is to provide us with a set of constraints which we then can embed within the ACT-R theory of higher-level cognition.

We have implemented the spotlight metaphor of visual attention where a variable-sized spotlight of attention can be moved across the visual field. When the spotlight fixates on an object, it's features can be recognized. Once recognized, the objects are then available as declarative structures, called chunks, in ACT-R's working memory and can receive higher-level processing. The following is a potential chunk encoding of the letter H:

object
isa H
left-vertical bar1
right-vertical bar2
horizontal bar3

We assume that before the recognition of the object, features (e.g., the bars) are available as part of an object but that the object itself is not recognized. In general, we assume that the system can respond to the appearance of a feature anywhere in the visual field and recognize the objects. However, it cannot respond to the conjunction of features that define a pattern until it has moved its attention to that part of the visual field and recognized the pattern of features. Thus, in order for the ACT-R theory of higher-level processing to "know" what is in its environment, it must move its attentional focus over the visual field. In ACT-R the calls for shift of attention are controlled by explicit firings of production rules. Consequently, it will take time to encode visual information and we are forced to honor the limited capacity of visual attention.

A basic assumption is that the process of recognizing a visual pattern from a set of features is identical to the process of categorizing an object given a set of features. Anderson and Matessa (1992) provide a rational analysis of how to perform such categorization without commitment to a particular cognitive architecture. That theory provides us with the mechanism for assigning a category (such as H) to a particular configuration of features. We have implemented this mechanism within the ACT-R for translating stimulus features into chunks like the above which can be processed by the higher-level production system,

The best way to illustrate how this theory functions is to describe how it functions in particular tasks. In the following section we will first describe how the theory would apply in modeling data from the classic Sperling Task which will give us some estimate of time to move visual attention. Then we will discuss an application of this to the subitizing task which shows how this time estimate plays out in a slightly more complex situation. We will then turn to discussing movement of visual attention in a higher-level task—solving equations.

Sperling Task

Sperling (1960) reported a classic study of visual attention. In the whole-report condition he presented subjects with brief presentations of visual arrays of letters (3 rows and 4 columns) and found that on average they could report back 4.3 letters. In the partial-report condition he gave subjects an auditory cue to identify which row they would have to report. Then he found that they were able to report 3.3 letters in that row. As he delayed the presentation of the auditory cue to 1 second after the visual presentation he found that subjects' recall fell to about 1.5 letters. Since subjects' recall at a second's delay fell to about a third of the whole report level, the obvious interpretation was that they were able to report as many items from the cued row as they happened to encode without the cue. This research has been interpreted as indicating that subjects have access to all the letters in a visual buffer but they have difficulty in reporting them before they decay. We will use .8 seconds as our estimate of the duration of that buffer in our simulation. We represented the letters in the visual array as sets of features grouped in unidentified objects. Depending on the situation, one of the following two productions would apply:

Encode Screen
IF one is encoding digits without a tone
and there is an object on the screen that
has not be attended
THEN move attention to that object

Encode Row
IF one is encoding digits and there is a tone
and a row corresponds to the tone
and there is an object in the row that
has not be attended
THEN move attention to that object

These productions call for attention to be moved to unattended objects. When the production moves attention to the location of that object, the letter would be recognized and a chunk created to encode it. This chunk creation also allows ACT-R to know it has attended to that object (and so avoid return visits). The actual recognition of the letter is done by the categorization component external to the productions. If no tone is presented, *encode-screen* will encode any letter in the array; whereas, if a tone is present, *encode-row* will encode letters in the cued row. Thus, the number of letters encoded is essentially equal to the number of productions that can fire in .8 seconds. After the visual array disappears, the simulation can report only those letters that had been encoded because only these have a chunk representation in working memory.

We get the just-over-four letters reported in the whole-report procedure (as found by Sperling) by setting the time per attention-switching production rule to .2 seconds. This was a mean time for a production to apply; we added a stochastic component to these times producing item-to-item variability in times. Because of this stochastic component the model averaged slightly over 4 letters reported in .8 seconds. To see this, suppose that the array was just available for .2 seconds and half of the times for the production were under .2 seconds and half were above. For those under, the model would get a second look and so encode a second letter (the assumption is that the letter is encoded as soon as attention fixates). For those over, there would be just one letter reported. So the average number reported would be 1.5 letters.

Performance in the partial report condition was in a certain sense suboptimal because attention would not switch as soon as the tone sounded but rather as soon as the next production fired after the tone sounded. Thus, if the tone sounded at .3 seconds and the next production fired at .4 seconds the subject would only have .4 (.8 - .4)

seconds to scan the appropriate row. Even when the tone occurred immediately at the offset of the array the first attentional fixation would be at a random location on the array and the second would be set to the target row. Sperling reports performance somewhat lower in the partial report condition than would be expected if the subject could report as many terms from the target row as the full array. Our simulation reproduces this effect in a parameter free way.

Subitizing

We have taken the 200 msec estimate of the time to switch visual attention from the Sperling task and used it to model a number of other tasks which involve deploying visual attention and we will describe here its application to subitizing (see the recent discussion by Simon, Cabrera, & Kliegl, 1993). Figure 1 illustrates the classic result obtained in this task where there is an increase in latency with number of digits to be identified. However, there is an apparent discontinuity in the increase with the slope being much shallower until 3 or 4 items and then getting much steeper. There is about a 50 msec slope until 3 or 4 items and approximately a 250 msec slope afterwards. Figure 1 also shows the results from the ACT-R simulation.

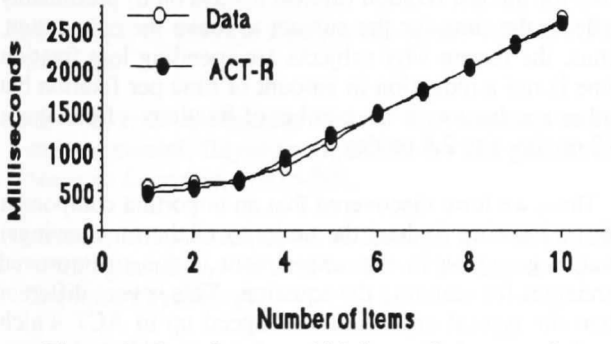

Figure 1: Data from a subitizing task compared to predictions of the ACT-R theory.

The basic organization of the model is to assume that there are special productions that recognize 1, 2, 3, and familiar configurations of larger number of objects (such as five on a die face) and that there is a production which can count single objects. This is the basic model of the subitizing task that has been proposed by researchers such as Mandler and Shebo (1982). The following are two of the productions used in modeling the task:

Recognize-Two
IF the goal is to count the objects starting from a count of 0
and there is an object at position 1
and there is an object at position 2
THEN move attention to their pattern and the count is 2.

Count-One
IF the goal is to count the objects starting from a count of m
and there is an object on the screen that has not be attended
and n is the successor of m
THEN move attention to that object on the screen and increment the count to n.

Faced with an array of objects, the largest pattern-recognition production (like *recognize-two* above) will apply and directly recognize the count of those objects. After that point it is necessary to add further objects into a running count and this is done by *count-one* above. While one could count multiple additional objects and add them into the existing count, the simpler model that we have implemented simply counts up by ones. The basic latency model for this task is one which takes the 200 msec from the Sperling task as the time for shift of attention and 50 msec as the time to match each non-goal element in a production rule beyond the first. The ACT-R model assumes that production firing increases with number of production condition elements. The basic latency will be determined by the productions above plus the time for response generation. Small arrays will be handled by productions like *recognize-two* with each object matched taking an additional 50 msec to match. The *count-one* production, which determines the per-element time (beyond 3) for larger arrays, is basically the production from the Sperling task plus a retrieval of the successor of the count. Each firing of it should take 250 msec, reflecting the 200 msec to shift attention plus 50 msec to retrieve the successor of the count.

Equation Solving

Figure 2 shows a screen image that we have been using in our research on equation solving. We have done extensive research on how subjects solve such equations and much more complex ones (Anderson, Reder, & Lebiere, submitted; Lebiere, Anderson, & Reder, 1994). This early work involved simulations which assumed that subjects were operating on an internal representation of the equation which came as an encoding of the visual presentation of the screen. More recently, we have been interested in modeling how subjects might actually go about encoding information from the screen. We have assumed that subjects first encode the symbols from the equation through a scanning process driven by productions like:

Encode-Symbol
IF the goal is to solve an algebra equation
and the leftmost unattended object is at a location
THEN move attention to that location.

This production embodies a left-to-right encoding strategy in which subjects encode each symbol from the equation and then are able to apply some procedure like:

Multiply-both-sides
IF the goal is to solve an algebra equation
and the equation is of the form X/C = D
THEN set a subgoal to multiply C by D.

However, in observing our own behavior solving this equation we noted that we often did not bother to scan the whole equation but rather focused on just the meaningful parts of the equation — the C and D in X/C = D. (This occurred in the context of an experiment where all the problems were division problems.) Therefore, we have begun a research program to study just how subjects do scan such equations.

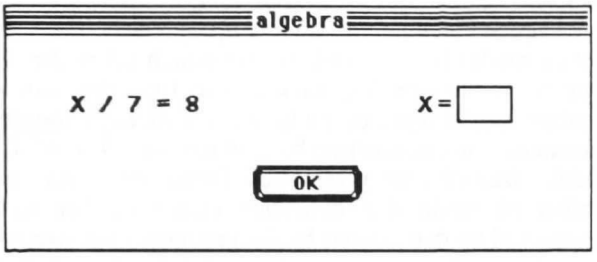

Figure 2

Our initial research has involved using a restricted interface like the one illustrated in Figure 3. Here the subject has a movable window with which to examine the equation. To move the window, the subject just moves the mouse. We analyzed subject movements into periods where they spend at least 200 msec. on a meaningful symbol of the equation. If we represent these simple equations as X/C = D, then the five meaningful regions are X, /,C,=, and D. The spacing and size of the window are such that only one region at a time can be fixated.

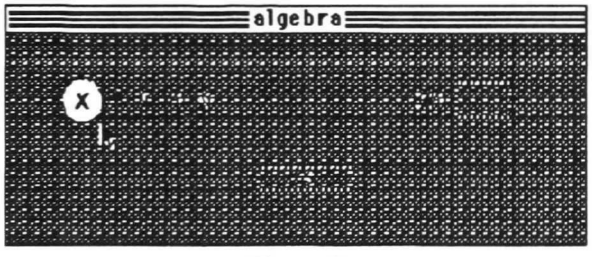

Figure 3

We found evidence for at least two scanning patterns. One involved a near-exhaaustive, linear scan of the equation in which the subject fixated the symbols in the sequence X, /, C, =, D. If we use this strict definition then subjects engage in this scanning pattern (with over 200 msec fixations on each symbol) on just 8% of the problems. However, if we use a more liberal scoring definition in which we require the subject to fixate C, D, and two of X, /, and D in any order with any number of repeat visits that percentage rises to 43%. The second strategy was indicated by the subject just visiting the C and D. This occurred 35% of the time. These percentages are for the first day of the experiment. Subjects were in the experiment for three days. By the third day the first strategy had dropped from 43% to 11% by the liberal scoring method while the second strategy had risen from 35% to 67%. Thus, there is a very definite shift over the course of the experiment to a more efficient scanning strategy.

This raises the issue of what the nature of the learning might be in this task. Subjects get much faster over the course of the three days, taking 4.06 seconds to solve the simple equations on day 1 but 2.66 seconds on day 3. We broke this time up into two components. There is the time spent when not fixating the equation and the time spent fixating the equation. The non-fixation time includes the time before the first equation fixation when the subject is perhaps organizing a strategy and the time after the last fixation when the subject is typing out the answer. This non-fixation time decreases from 1.84 seconds on day 1 to 1.34 seconds on day 3. The fixation time decreases from 2.22 seconds to 1.32 seconds. We analyzed the fixation time into time per fixation and separated these out into fixations before the last and fixations after the last. The pre-last fixations appear to be relatively constant and average .42 seconds on day 1 and .36 seconds on day 3. So there is little speed up in the duration of these fixations. The last fixation takes much longer—.94 seconds on day 1 and .92 seconds on day 3. The longer time for the last fixation (almost always on d) presumably reflects the time for the subject to make the calculation. Thus, the reason why subjects are spending less fixation time is not a reduction in amount of time per fixation but rather a reduction in the number of fixations—from about 4.2 on day 1 to 2.8 on day 2

Thus, we have discovered that an important component of the learning (indeed the majority of the time savings) that is going on in this experiment is due to improved strategies for scanning the equation. This is very different than the typical explanation of speed up in ACT which attributes it either to stronger and more rapid productions or to compositions of existing productions (Anderson, 1987). This serves to illustrate the important contribution that study of visual attention can make to our understanding of the nature of learning. This research indicates that an important component ot skill development is learning where critical information is to be found in the visual interface. The ACT-R theory does have strategy learning mechanisms which can model the transition between strategies (Anderson, 1993; Lovett & Anderson, in press) and we are currently in the process of trying to model this transition.

Concluding Remarks

In the introduction we described two motivations for developing a theory of visual processing in ACT-R. One was to model the information-processing limitations in accessing information from the screen. This paper has been mainly devoted to describing that—showing how we can model classic attentional paradigms and gain insight into the performance and improvement of complex

cognitive skills. The other motivation was to eliminate magical degrees of freedom in going from a description of an experiment, to a cognitive model of how it is performed. We have accomplished this also. The same experimental software that runs subjects interacts with the ACT-R system.[1] We have developed the experiment-running system that can be "toggled" so that it will either administer an experiment to a real subject or interact with the ACT-R simulation. When it is toggled to interact with ACT-R, ACT-R can "see" the screen in terms of this feature representation and the experimental program will read key presses, mouse movements, and mouse clicks issued by ACT-R. Thus, it becomes possible for anyone to build a simulation to interact with the same experiment-running software that subjects interact with (provided that software is written in LISP on the Macintosh).

References

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review, 94*, 192-210.

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

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

Anderson, J. R. & Matessa, M. (1992). Explorations of an incremental, Bayesian algorithm for categorization. *Machine Learning, 9*, 275-308.

Anderson, J. R., Reder, L. M. & Lebiere, C. (submitted). Working memory: Activation limitations on retrieval. *Cognitive Psychology*.

Kieras, D. E. & Meyer, D. E. (1994). The EPIC architecture for modeling human information-processing and performance: A brief introduction. EPIC Report No 1. (TR-94/ONR-EPIC-1). University of Michigan.

Lovett, M. C. & Anderson, J. R. (in press). History of success and current context in problem solving: Combined influences on operator selection. *Cognitive Psychology*.

[1] We have taken advantage of standards for Macintosh LISP dialogue windows to create a default visual interface module which represents alphanumeric characters in terms of a standard set of features (those used in the McClelland & Rumelhart, 1981, modeling), and encodes all other elements in the window in terms of their default Macintosh encodings. It is possible for users to change the feature representation according to their preferences, but the default provides a first-order representation from which everyone can work

Lebiere, C., Anderson, J. R., & Reder, L. M. (1994). Error modeling in the ACT-R production system. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, 555-559. Hillsdale, NJ: Erlbaum.

Mandler, G. & Shebo, B. J. (1982). Subitizing: An analysis of its component processes. *Journal of Experimental Psychology: General, 111*, 1-22.

McClelland, J. L. & Rumelhart, D. E. (1981). An interactive model of context effects in letter perception: I. An account of basic findings. *Psychological Review, 88*, 375-407.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology, 32*, 3-25.

Simon, T., Cabrera, A., & Kliegl, R. (1994). A new approach to the study of subitizing as distinct enumeration processing. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, 929-934. Hillsdale, NJ: Erlbaum.

Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs, 74*, 1-29.

Treisman, A. M. & Sato, S. (1990). Conjunction search revisited. *Journal of Experimental Psychology: Human Perception and Performance, 16*, 459-478.

Wiesmeyer, M. D. (1992). An operator-based model of covert visual attention. PhD Thesis, The University of Michigan, Ann Arbor.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review, 1*, 202-238.