Title
Dissociating Performance from Learning: An Empirical Evaluation of a Computational Model

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 18(0)

Authors
Vera, Alonso H.
Lewis, Richard L.

Publication Date
1996

Peer reviewed

Dissociating Performance from Learning: An Empirical Evaluation of a Computational Model

Alonso H. Vera
Department of Psychology
The University of Hong Kong
Pokfulam Road
Hong Kong
vera@hkucc.hku.hk

Richard L. Lewis
Department of Computer and Information Science
Ohio State University
2015 Neil Avenue Mall
Columbus, OH 43210
rick@cis.ohio-state.edu

Abstract

This paper presents a follow-up to the ATM-Soar models presented at 1993 Meeting of the Cognitive Science Society and the CHI 1994 Research Symposium. The original work described the use of the Soar cognitive architecture to simulate user learning with different ATM interfaces. In particular, it focused on the relative effects of interface instructions (e.g., "Insert card into slot") and perceptual attentional cues (e.g., a flashing area around the card slot) on learning and performance. The study described here involves getting human data on the same tasks to test the predictions of the computational models. The ATM task is simulated on a PC in order to contrast three types of interface conditions: just instructions, instructions plus flashing, and just flashing. Subjects must insert a bank card, check the account balance, and withdraw money. They are asked to repeat the task four times so that the effects of training on performance and learning can be observed. The data suggests that subjects learn to perform the task faster with attentional attractors, as the Soar model predicted. More interestingly, the Soar model also predicted that people would do better *without* instructions when there are attentional attractors. This prediction was supported as well.

In recent years, we have seen the rise of a number of AI cognitive architectures (e.g., Soar (Newell, 1990) and ACT-R (Anderson, 1993)) which attempt to provide unified theories of psychological phenomena. We have also seen the growing use of these architectures in the field of Human-Computer Interaction (HCI). Perhaps the primary scientific motivation for studying HCI is to provide a testing ground for our computational models of cognition, particularly those that describe learning and performance in interactive tasks. This paper reports on a set of studies currently in progress to empirically evaluate the predictions of a Soar model originally presented at the 1993 Meeting of the Cognitive Science Society (Vera, Lewis and Lerch, 1993).

In the 1993 paper, we described a Soar model that simulated a user learning to interact with different ATM interfaces. In particular, we focused on the relative effects of interface instructions (e.g., "Insert card into slot") and perceptual attentional cues (e.g., flashing area around the card slot) on learning rate for the task. In Soar, the number of memory chunks formed was taken as the measure of ease or difficulty of learning in the different conditions. One of the basic outcomes of the ATM-Soar model was that better interfaces lead to less learning during task performance. This result needs to be evaluated empirically. The follow-up study presented here involves getting human data on these tasks.

The ATM-Soar Models

There were two related goals in building the ATM-Soar model. The first goal was to answer a set of questions about the cognitive processes and representations of the user in the ATM scenario. In particular, we were interested in how the task was mentally represented and accomplished, and how that representation evolved as a function of learning. A good cognitive model should answer the following questions:

- How is behavior initially guided in the task?
- What determines the sequence of actions taken by the user?
- What exactly is learned as a result of performing the task?
- How does the learning affect performance on later trials?
- What constitutes expert or optimal performance on this task?

The second and related goal of the modeling was to understand how aspects of the interface affect performance and learning, and to use that understanding to suggest changes in the interface design. The cognitive model should help answer the following questions:

- What computational, functional, and knowledge demands does the interface and task place on the user?
- How does the interface design affect learning?
- How can the interface be changed to decrease both the time to accomplish the task, and the time required to reach expert performance?

The Soar modeling effort was primarily focused on cognitive skill acquisition and the cognitive demands of the task. We were not concerned with details of motor behavior, or the interleaving of motor execution with cognitive processing. We did not model behavior at the key-stroke level (e.g., John, Vera & Newell, 1994), nor did we present detailed perceptual models (e.g., Wiesmeyer, 1992). All of these approaches are important and useful, and complement the approach we have taken here.

Soar, as an architectural theory, brings with it independently motivated principles of task performance and acquisition (Newell, 1990). A number of other architectures could have also been adopted (e.g., ACT-R); the minimum requirements for our present purposes are that the architecture specifies exactly how goal-directed task behavior unfolds, and how that behavior can change over time as a result of some kind of learning.

Adopting Soar as the underlying theory has a number of important implications for our task that are apparent even before specifying a detailed model. The representation of the task must consist of a set of independent associations in long term memory, cued by the contents of working memory. Task behavior is not fixed in advance by a plan structure or rigid program in memory; rather, behavior is a function of whatever knowledge is immediately cued and assembled at the time of action. Finally, because chunking is an experiential learning mechanism, the task must be learned by doing the task. Although prior preparation (e.g., instructions) may be helpful, there is no substitute for practice.

Cognitive architectures are programmable -- that is their primary functional feature. An architecture without content will not yield behavior. Behavior is a function of the fixed architectural mechanisms, the contents of the memories, and the current situation. This means that to develop a complete model, the theorist must posit the knowledge that a subject brings to the task. (Reducing the degrees of freedom in this step is an important methodological issue for cognitive science; see Newell, 1990; and Lewis, Newell & Polk, 1989, for more discussion).

How do we specify the content of the ATM models? The guiding principle is to make plausible assumptions about the knowledge and skills that a user will bring to the ATM task for the first time. All users bring to the task a set of general cognitive capabilities such as language comprehension and the ability to direct attention to different regions in space. These capabilities are functionally required for the task, but the details of their implementation are not our concern here.

The models developed posited a set of abstract functional capabilities that were realized in the Soar model by a set of operators that served as place holders for the more detailed mechanisms. In particular, the models assume pre-existing operators that comprehended language, shifted attention and intended motor behavior. While this did not permit us to explore the effects of the interface on the internal structure of these operations, it permitted asking critical questions about how these given cognitive functions are deployed to accomplish the task.

In addition to these general capabilities, we must posit some task-specific knowledge as well. Although it would be possible to simply posit expert-level memory structures, we are interested in how these structures arise. Thus, we make fairly minimal assumptions about the knowledge a user brings to the task initially:

1. Knowledge of task objects. The user knows he has an account with a balance, knows he has a plastic card that is required to operate the device, and knows he has a personal identification number (and knows what it is).
2. Knowledge of physical devices. The user knows how to push buttons and insert cards into slots, and furthermore can make some simple associations between aspects of the device and possible task-related actions (for example, the slot may be good for inserting the card, the numeric keypad may be good for specifying dollar amounts or PINs).
3. Minimal task strategy. The user does just what is needed to accomplish the task, and no more. We assume that the basic strategy guiding behavior is simply looking around the device for cues about what to do next, which may take the form of explicit task instructions. The user's goal is not to learn how to use the machine, but to get the account balance (or whatever) and leave.

The basic principles of the ATM-Soar model have already been described in Vera et al., (1993). An extension to that model, presented at a Research Symposium following CHI'94, showed that using perceptual cues in the interface to attract the model's attention to the relevant location greatly reduced the number of chunks built during learning. In other words, much of what the Soar model learned in the original version was a consequence of having to search around the interface in order to find the next relevant information. The second model assumed that attention could be drawn to the relevant part of the interface with perceptual cues.

This second model achieved the same level of performance as the original, but learned much less because it did not have to memorize the sequence of places in the interface to which it needed to attend. The argument presented in this paper is that the same is basically true for human users. To the extent that the interface has to be searched to find the next relevant action to execute, more learning is required in order to improve performance. If, on the other hand, searching is reduced or removed completely by having attention drawn to the relevant part of the interface, then performance improves, but the amount of information the user has to learn should not increase. Here, we present a study that explores these predictions by having subjects perform the ATM task on simulated interfaces with and without perceptual cues.

The ATM Study

The Soar model predicts that attentional-cues should make a big difference in the performance of people using ATMs without instructions. The attentional cues should speed up the process of achieving "expert-level" performance. Moreover, this should be attained without a concomitant increase in learning.

Although it may seem somewhat counterintuitive, what is being suggested is that perceptual cues will lead to a steeper performance improvement curve (i.e., subjects will get better faster) while less actual learning is going on. This is because the cues will guide attention without adding cognitive processing that would increase learning. This raises an important distinction, since learning is often measured in terms of improvements in performance. As the Soar model suggests, the relation between performance and amount learned (number of new chunks in the Soar model) depends strongly on the interface. A user who has learned many new things about one interface may still perform more poorly than a user who has learned little about a different interface. Furthermore, if the Soar model is correct, it leads to another counter-intuitive prediction that, following a number of training trials, performance with the attention cues will actually be faster without the instructions than with the instructions, because the instructions just get in the way at this point. Possible explanations for this are discussed in the Results section.

A flashing border around the relevant interface object was selected as the perceptual cue to be used in these studies although a number of other alternatives were available. Other variables that might have the property of cueing perception in 2-D environments are things like changes in shape and size, appearance/disappearance of objects, movement of objects, and coordinated movement of more than one object. There are also other candidates such as color changes, sound from a particular part of the interface, and so on, but these are not likely to be helpful given the typical physical locations of ATMs in the real world. These latter cues have often been used in interfaces since they tend to be the easiest and most obvious way to attract attention.

The ability of the perceptual cues to attract attention was measured in terms of the time it took to achieve the next task action (i.e., time to the next correct mouse click). Some recent work has treated these sorts of perceptual cues as "affordances", in the Gibsonian (1979) sense that they directly cue action (e.g., Howes & Young, in press). This is not the idea here. The only thing that flashing does is attract the user's attention -- action is generated by independent cognitive processing of task goals and current conditions. The effect of attention cues versus instructions is thus measured in terms of reaction time. This is actually a measure of Attention + Cognition (decide what to do) + Motor (do it). Assuming that the motor behavior itself does not change significantly across the conditions, the difference between the two conditions is due to differences in Attention + Cognition; that is, differences in the time required to decide where and what to initiate next.

Method

Subjects were 96 undergraduate students from The University of Hong Kong. Simulation interfaces were built and run on a 486 PC platform. A high-resolution digitized photograph of an ATM was used to generate the look of the simulation. All of the functional features of the interface worked exactly like those of a real ATM. Subjects interacted with the interface using a mouse. They could drag objects like the bank card, click on buttons, drag money from the dispensing slot, and so on (see Figure 1).

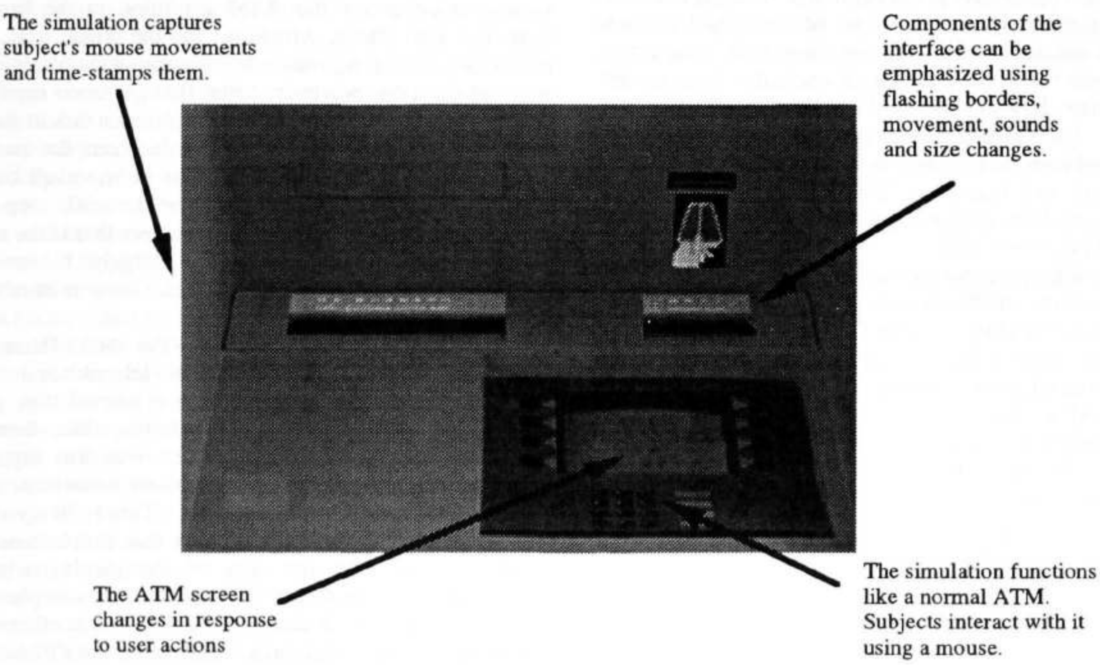

Figure 1. Characteristics of the ATM Interface Simulation

There were two interface conditions. First, a normal ATM interface was used. All the functions of a common ATM were fully reproduced. Subjects' task was to insert the bank card into the slot, type in their Personal Identification Number (PIN), select a function (get their account balance), select an account to check (from two possibilities: checking and savings), select another function (withdrawal), select an account to debit, enter the amount, and remove the money. They repeated this task four times.

In order to address the fact that subjects had previous experience with ATM machines, the task was also modified in a separate condition so that subjects were not performing an ATM transaction but instead choosing a new telephone card number. The second interface condition was therefore an invented "Phone Machine" which looked just like an ATM except that we replaced the bank logo with a Hong Kong Telecom logo and the ATM card with a phone card. The functional aspects of the interface remained unchanged. The (made-up) functions of the Phone Machine were explained to subjects at the outset. They were told that, among other things, they could settle their accounts with the phone company, check how much they owed, change their personal phone code, and so on. The actual task they performed was to first check how much money they owed the phone company and then change their phone code. The individual steps required them to insert the phone card into the slot, type in their phone code, select a function (get their account balance), select a billing option (from two possibilities: pay by check or charge to credit card), select another function (change secret phone code), select a code to change, enter the new code, and remove a statement. Subjects did this four times.

There were 16 subjects per condition for the ATM task and 16 per condition for the Phone Machine task. Each subject was asked to carry out the same task four times because it was the same number of trials the Soar model required to learn how to perform the task without using the instructions. Each trial was separated by the same distracter task where subjects were asked to count backwards by 17's from 1000 for two minutes. This was done in order to prevent subjects from rehearsing the task once they realized they were doing it repeatedly.

The main manipulation of this study involved varying perceptual aspects of the interface to attract attention to specific areas of the display. The functionally relevant part of the display had a flashing surrounding border. The display objects affected by the flashing were the card slot, the numerical keypad, the information screen, and the buttons around the screen. In one condition subjects saw instructions, but no flashing; in a second condition, they saw both instructions and flashing; and, in the third condition, just flashing with no instructions. There were therefore 6 experimental groups in a 2 task (ATM vs. Phone Machine) X 3 interface (instruction, no flashing vs. instruction and flashing vs. no-instruction, flashing) design. The design was between-subjects design and each subject saw only one task and one interface type.

Results

In order to compare subjects' performance across the three conditions, one component of the task was chosen. The time from the screen change following insertion of the card until the first number of the PIN was clicked was measured. The following comparisons are based on performance on this measure during fourth training trial across subjects. Performance was fastest in the condition with flashing but no instructions, as anticipated. T-tests showed that on the fourth trial, performance was significantly faster in the condition with flashing but no instructions (1.83 sec) than in the condition with flashing as well as instructions (2.25 sec), $t(93)=2.25$, $p<.05$. The performance difference between the condition with flashing but no instructions (1.83 sec) and the condition with instructions but no flashing (2.13 sec) was close to significance at the .1 level. The analyses suggest that people's final performance is faster when there are no instructions present. This follows from the hypothesis that instructions demand cognitive resource even when the user already knows how to perform the task.

The task manipulation (ATM vs. Phone Machine) yielded no significant performance differences across conditions and trials. There are at least two possible explanations for this. The Phone Machine interface may not have been sufficiently dissimilar to the ATM's in terms of its physical and functional characteristics. Alternatively, top-down knowledge from the familiar ATM task may have transferred quite easily to the novel Phone Machine task. If the former is the case, then subjects' performance on the two tasks should have been quite similar from the first trial onward. If the latter is true, then performance should be somewhat better for the ATM condition in the first trial than for the Phone Machine in the first trial, with performance evening out over the subsequent trials. This was not the case however, since there was no significant difference between the performances on each task in the first trial, suggesting that the similarity between the two task conditions was the main factor. This is important because it suggests that tasks variables override top-down knowledge from the beginning. It is clear that if the effects of these manipulations were due largely to top-down knowledge, results regarding interface characteristics would lose some of their meaning.

In summary, subjects perform the task faster with attentional attractors, as the Soar model predicted. More interestingly, the Soar model also predicted that people would do better without instructions when there are attentional attractors. This prediction was also supported. Performance on the fourth trial is slower when instructions are present than when they are not. This is likely due to two independent factors. The first is that instructions draw attention away from the area of the display which is functionally relevant to the next task. For example, ATM user's attention may be drawn toward the instruction screen rather than to the numerical keypad when the PIN needs to be entered. The second reason is that, once attended, the text on the screen is processed automatically (see, e.g., Fodor (1983) for a discussion of this mandatory quality of input systems, and Newell (1990) and Lewis (1996) for a discussion of how Soar accounts for such modularity effects) and consequently uses additional cognitive resources.

Discussion

Until recently, the consensus in cognitive science was that processes such as memory, problem-solving, categorization, and causal inference were non-optimal because they did not perform maximally in many conditions. This is because these conditions happen to be conditions that do not exist in the real world. Rational analysis (Anderson, 1990) suggests that information in the external world is structured such that our cognitive systems can take maximal advantage of it. Our systems are optimally tuned to information out in the real world because they evolved to predict not just any arbitrary set of external conditions but those that actually hold in this world.

This sort of approach would suggest that structuring the external world (an interface, in the case of HCI) such that users can interact optimally with it is not a matter of turning to concepts such as "affordances" to solve the problem. There is currently no evidence that anything acts like an affordance (in the true Gibsonian sense) in computer interfaces. Although most of today's GUI's use buttons, sliders, and so on, there is little reason to believe that these images are directly cueing action in any way. It is even doubtful whether real world buttons (e.g., in an elevator) *afford* pressing.

The approach of the study here was to enlist low-level perceptual cues to guide attention to relevant parts of the interface. This is based on our computational model's prediction that the critical time bottleneck in this task comes from searching the interface for relevant information. This approach is quite different from attempts to improve performance by redesigning aspects of the interface so that they directly cue or afford the relevant action. While it may be the case that certain object designs are better cues to relevant actions than others, such cues only solve part of the problem. In particular, they do not provide a way to reduce the time spent searching the interface for something relevant because they do not function by explicitly drawing attention. They are important insofar as they facilitate the evaluation of relevancy, and guide action in service of goals once the relevant part of the interface is attended. Indeed, as mentioned earlier, associations from aspects of the interface to possible actions was an important part of the Soar models' initial knowledge.

In short, the two approaches complement each other. The goals of interface design might be best served by working on both problems: guiding attention with low-level perceptual cues, and using object designs that provide good cues for the next set of possible actions. The present Soar models suggest that in certain tasks, guiding attention to reduce search may be the most important factor. Increased search time is detrimental not only to performance, but also to learning because it forces users to learn more than they have to.

The study presented here looked at the effects of instructions and perceptual cues on *performance*; it did not evaluate the relative effects on *learning*. The next set of studies will attempt to separate the learning components of the task. It seems clear from the results already available that the continued presence of instructions over trials lowers performance. It is also clear that perceptual cues improve it. Furthermore, performance is significantly impaired when instructions are removed. What cannot be determined from the present study is whether performance will deteriorate relatively more when flashing is removed (i.e., when subjects are trained on trials with flashing and then tested on trials without it). The Soar model predicts a greater drop in performance when flashing is removed than when instructions are removed because the flashing condition leads to fewer chunks being built (i.e., less being learned).

The studies presently being conducted present subjects with four training trials, like the study described here, plus three test trials where the instruction / flashing conditions are varied. In particular we are interested in seeing what happens to performance when subjects are trained with flashing but no instructions, and then tested with no flashing or instructions. We expect that performance will deteriorate more in this condition and than in a condition where they go from instructions with no flashing to no instructions and no flashing. This is because, if the model's prediction that flashing leads to better performance but less learning is correct, then performance should fall off steeply when flashing is no longer available since subjects will have learned very little about the task. The goal is therefore to demonstrate that, as predicted by the ATM-Soar model, a better interface is one that requires less learning in order to achieve better performance.

Acknowledgments

Work on the research reported in this paper was supported by CRCG grant 337/014/0021.

References

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

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

Fodor, J. A. (1983) *Modularity of mind.* Cambridge, MA: MIT Press.

Gibson, J. J. (1979) *The ecological approach to visual perception.* Boston, MA: Houghton-Mifflin.

Howes, A. and Young, R. M. (in press). Learning consistent, interactive and meaningful task-action mappings: A computational model. *Cognitive Science.*

John, B. E., Vera, A. H., and Newell, A. (1994). Towards real time GOMS: A model of expert behavior in a highly interactive task. *Behaviour and Information Technology,* **13**, 4, pp. 255-267.

Lewis, R. L. (1996). Architecture matters: What Soar has to say about modularity. In Steier, D. and Mitchell, T., (Eds.), *Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell.* Hillsdale, NJ: Erlbaum.

Lewis, R. L., Newell, A. and Polk, T. A. (1989) Toward a Soar theory of taking instructions for immediate reasoning tasks. In P. S. Rosenbloom, J. E. Laird, and A. Newell (Eds.), *The Soar Papers: Research on Integrated Intelligence,* Cambridge, MA: MIT Press.

Newell, A. (1990) *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Vera, A. H., Lewis, R. L., and Lerch, F. J. (1993). Situated decision-making and recognition-based Learning: Applying symbolic Theories to interactive tasks. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.* Boulder, CO.

Wiesmeyer, M. D. (1992). An operator-based model of human covert visual attention (Tech Report CSE-TR-123-92). Doctoral Dissertation. Ann Arbor: The University of Michigan, Department of Electrical Engineering and Computer Science.