

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

An ACT-R List Learning Representation for Training Prediction

Permalink

<https://escholarship.org/uc/item/625367wk>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 32(32)

ISSN

1069-7977

Author

Matessa, Michael

Publication Date

2010

Peer reviewed

An ACT-R List Learning Representation for Training Prediction

Michael Matessa (mmatessa@alionscience.com)

Alion Science and Technology
6404 Cooper Street
Felton, CA 95018 USA

Abstract

This paper presents a representation of training based on an ACT-R model of list learning. The benefit of the list model representation for making training predictions can be seen in the accurate a priori predictions of trials to mastery given the number of task steps. The benefit of using accurate step times can be seen in the even more accurate post-hoc model results.

Keywords: Training; prediction; list length; ACT-R.

Introduction

Numerous studies have documented operational and training problems with the modern autoflight systems, in particular the flight management system (FMS) and its pilot interface, the control display unit (CDU). During the last few years, more attention has been given to the limitations of current autoflight training methods. Many studies have concluded that current training programs are inadequate in both depth and breadth of coverage of FMS functions (Air Transport Association, 1999; BASI, 1998; FAA Human Factors Team, 1996).

Matessa and Polson (2006) proposed that the inadequacies of the programs are due to airline training practices that encourage pilots to master FMS programming tasks by memorizing lists of actions, one list for each task. Treating FMS programming skills as lists of actions can interfere with acquisition of robust and flexible skills. This hypothesis of the negative consequence of list-based representation was validated by Taatgen, Huss, and Anderson (2008), who show poorer performance for list-based representation compared to a stimulus-based representation.

This paper extends the table-based training time predictions of Matessa and Polson (2006) by presenting a computational model that represents procedure training as list learning. The model is meant to describe training programs where to-be-learned procedures are formally trained, and trainees must demonstrate mastery before they can go on to more advanced, on-the-job training. Airline transition training programs are examples of this paradigm. The model takes as input the number of steps in a procedure and the time per step, and it generates estimates of the training time required to master the procedure. Predictions of the model are compared to human data and show the benefit of the number-of-steps and step-time parameters.

Model

Novice pilots lack an organizing schema for memorizing lists of actions and so the actions are effectively represented as nonsense syllables (Matessa & Polson, 2006). Therefore, the list model does not represent the actual information to be learned, but instead as an engineering approximation represents the training as learning a list of random digits. The model is motivated by the table-based list model of Matessa and Polson (2006), but is implemented in the ACT-R cognitive architecture (Anderson, 2007).

Table-Based List Model

The following description from Matessa and Polson (2006) shows how procedure learning can be represented as list learning, and a table-based prediction of training time can be created based on procedure length. A representation of a task must encode both item (actions and parameters) and order information. Anderson, Bothell, Lebiere, and Matessa (1998) assumed that item and order information is encoded in a hierarchical retrieval structure incorporated in their ACT-R model of serial list learning shown in Figure 1. The order information is encoded in a hierarchically organized collection of chunks. The terminal nodes of this retrieval structure represent the item information. The model assumes that pilots transitioning to their first FMS-equipped aircraft master a cockpit procedure by memorizing a serial list of declarative representations of individual actions or summaries of subsequences of actions. It is assumed that each of these attempts to learn the list is analogous to a test-study trial in a serial recall experiment.

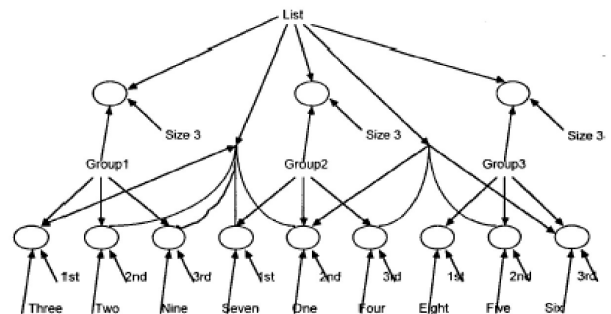


Figure 1: The List Model representation for a list of nine digits (from Anderson et al., 1998).

An interpretive process uses the list to perform the procedure. This process incorporates the knowledge necessary to understand each step description and to execute actions necessary to perform each step. Thus, an item such as “Press the LEGS key” would generate the actions required to locate the Legs key on the CDU keyboard and press it. A parameter such as a waypoint identifier would be represented in working memory as a sequence of letters. The interpretative process would generate the keystrokes necessary to enter the identifier into the scratch pad.

The list actions representation is a consequence of pilots’ decisions to treat the task of mastering FMS procedures as learning serial lists of actions. The retrieval structure shown in Figure 1 is generated by processes that adults use to memorize any arbitrary serial list of items. It is assumed that a novice representation of a FMS procedure with nine actions would be represented by replacing the terminal-node chunks with chunks representing individual actions in the procedure. The retrieval structure only encodes order information and supports access to the chunks representing individual actions. The groupings of the actions imposed by this structure have no relationship to the underlying task structure. Because these retrieval structures are unique to each task, they block transfer of training.

The following figure is a possible list describing an FMS procedure for the Boeing 777 for responding to the following hold clearance that would be generated by a pilot with limited glass-cockpit experience.

“NASA 1: Hold west of Haden on the 270 degree radial. Right turns. 10 mile legs. Expect further clearance at 2130 z.”

1. Press HOLD Function/Mode Key.
2. Press LSK 6L, if a holding pattern is in the route.
3. Line select waypoint identifier for Haden to scratchpad.
4. Press LKS 6L.
5. Enter the quadrant and the radial, W/270.
6. Press LSK 2L.
7. Enter the turn direction into the scratchpad, R.
8. Press LSK 3L.
9. Enter the leg distance into the scratchpad, 10.
10. Press LSK 5L.
11. Enter expect further clearance time, 2130.
12. Press LSK 3R.
13. Verify the resulting holding pattern on the ND.
14. Press EXECUTE.

Figure 2: A possible novice representation of a FMS procedure for responding to a Hold clearance.

This probably looks like a list of nonsense syllables to you, as it does to novice pilots. Pilots do not receive an explicit instruction on how to encode FMS procedures in memory early in training and lack organizing schemas that would help in memorizing instructions. Catrambone (1995) has shown that novices tend to describe problem solutions in terms of actions used to solve the problem. In the case of

FMS programming skills, this process leads to long lists that are very difficult to memorize.

The list shown in Figure 2 has undesirable properties and would be difficult to memorize. It is long—14 items—and it is organized as a linear sequence of actions that cannot be directly stored in memory (Anderson, et al., 1998). Some kind of idiosyncratic organization would have to be imposed on it to break it up into sublists before it could be successfully memorized. Furthermore, the representation of the procedure for programming a hold shown in Figure 2 is specific to a particular clearance. It would be relatively easy to generalize this representation to clearances with identical parameters but with different values. However, generalizing this procedure to cover the entry of any hold clearance requires numerous nontrivial inferences.

The Savings Paradigm The list model assumes that learning a FMS procedure is analogous to memorizing serial lists of nonsense syllables for a pilot with limited FMS experience. Training times can be estimated using results of an experimental paradigm initially developed by Ebbinghaus (1888/1913, Chapter 8). On the first day of the experiment, participants learn a serial list of items to a criterion of mastery of one perfect recitation of the list. Performance is measured as the number of trials to mastery. Participants return to the laboratory 24 hours later and relearn the list to the same criterion of mastery. Training stops on the first day that participants perform perfectly on the first presentation of the list after a 24-hour retention interval.

Table-based Prediction Matessa and Polson (2006) developed a table that presents the number of retentions on each successive day and the number of days of training required to be able recall a list perfectly after 24 hours. The numbers in the table were derived by synthesizing the results of several experiments from the list-learning literature starting with the data from Ebbinghaus (1885/1913, Chapter 8). The numbers are extrapolations generated by fitting power functions to Ebbinghaus’s results and then adjusting them to account for the fact that he used a very rapid presentation rate.

Training time is estimated by calculating the amount of time it would take to administer N repetitions of a procedure of length L during one session in a fixed-base or full-motion simulator. The model’s description of the training processes has three time parameters: session setup time (SST), repetition setup time (RST), and step time (ST). SST is the time required to set up a simulator to begin training a specific procedure. RST is the time required to set up the simulator for the next repetition, and ST is the time required for a trainee to perform a step and receive feedback from the instructor if necessary. These values are then summed over days to generate a training-time prediction for a given procedure.

The time devoted to training a procedure on one day = $SST + N \cdot RST + N \cdot L \cdot ST$.

The values for N, the number of repetitions on a day, are taken from the table. Values for SST and RST were set to 120 seconds, and ST was set to 5 seconds. Current fixed-based and full-motion simulators were found to be ill-suited to this kind of training; they are designed to simulate the execution of complete missions.

Numerous studies have shown that PC-based, part-task simulators can be used successfully to train skills such as performing FMS procedures (e.g., Salas, Bowers, and Prince, 1998; Salas, Bowers, and Rhodenizer, 1998; and Polson, Irving, and Irving, 1994). The lesson planners incorporated into commercially developed simulators can be programmed to deliver the necessary repetitions while minimizing the SST and RST (Aerosim Technologies, www.aerosim.com; Tricom Technologies, www.tricom-tech.com/products.htm; CAE, www.Cae.com; and Wicat, www.wicat.com). Use of such a trainer was modeled by reducing the values of SST and RST to 5 seconds.

ACT-R List Model

This paper presents a computational list model developed in the ACT-R cognitive architecture (Anderson, 2007). ACT-R includes a subsymbolic level of representation where facts have an activation attribute which influences their probability of retrieval and the time it takes to retrieve them. The activation A_i of a chunk i is computed from two components – the base-level and a context component. The base-level activation B_i reflects the recency and frequency of practice of the chunk. The equation describing learning of base-level activation for a chunk i is

$$B_i = \ln\left(\sum_{j=1}^n t_j^{-d}\right)$$

where n is the number of presentations for chunk i , t_j is the time since the j th presentation, and d is the decay parameter. The equation for the activation A_i of a chunk i including context is defined as:

$$A_i = B_i + \sum_j W_j S_{ji}$$

where the base-level activation B_i reflects the recency and frequency of practice of the chunk as described above. The elements j in the sum are the chunks which are in the slots of the chunk in module k . W_{kj} is the amount of activation from source j in module k . The strength of association, S_{ji} , between two chunks is 0 if chunk j is not in a slot of chunk i or is not itself chunk j . Otherwise it is set using the following equation:

$$S_{ji} = S - \ln(m)$$

Built into this equation is the prediction of a fan effect (Anderson, 1974) in that the more things associated to j the less likely any of them will be, on average, in the presence of j . That is, if there are m elements associated to j their average probability will be $1/m$.

The current model is an ACT-R 6.0 model based on the ACT-R 4.0 list learning model developed by Anderson et al. (1998) and can account for phenomena such as length and serial position effects. Figure 3 plots the probability of correctly recalling a digit in position as a function of serial position in input. There is considerable variation in recall of items both as a function of list length and input position. These variations are predicted by the model as a reflection of the changes in activations of the elements being retrieved. These activations increase with rehearsal (base-level activation), decrease with time (base-level activation), and decrease with list length (associative activation). As the list is longer, there will be greater interference because there will be more associations from the list element and less associative activation to any member of the list.

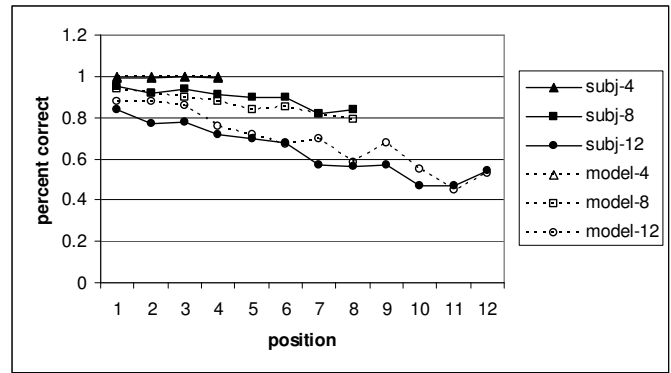


Figure 3: List model showing length and serial position effects.

In order to approximate training, the current model differs from the Anderson et al. (1998) model by not implementing its rehearsal strategy. In this way, presentation rate represents task step time (ST). As a consequence, longer presentation rates produce poorer performance, in contrast to findings from studies that allow rehearsal.

The model also uses the Pavlik and Anderson (2005) version of memory decay that accounts for spacing effects. They developed an equation in which decay for the i th presentation, d_i , is a function of the activation at the time it occurs instead of at the lag. This implies that higher activation at the time of a trial will result in the gains from that trial decaying more quickly. On the other hand, if activation is low, decay will proceed more slowly. Specifically, they propose

$$d_i(m_{i-1}) = ce^{m_{i-1}} + a$$

to specify how the decay rate, d_i , is calculated for the i th presentation of an item as a function of the activation m_{i-1} at the time the presentation occurred, with

$$m_n(t_1 \dots t_n) = \ln\left(\sum_{i=1}^n t_i^{-d_i}\right)$$

showing how the activation m_n after n presentations depends on the decay rates, d_i s, for the past trials.

These equations result in a steady decrease in the long-run retention benefit for additional presentations in a sequence of closely spaced presentations. As spacing gets wider in such a sequence, activation has time to decrease between presentations; decay is then lower for new presentations, and long-run effects do not decrease as much.

The model is run inside code that simulates the savings paradigm in order to determine trials to mastery. The model uses the same parameters as Anderson et al. (1998) except that the rate of presentation (representing step time) and repetition setup time are both set to 5 seconds, as in Matessa and Polson (2006). The activation retrieval threshold is set to -0.85 in order to match the predictions of the trials to mastery table found in Matessa and Polson (2006).

Experiment

In order to gather data for an experimental interface, Boeing conducted experiments with a PC-based, part-task simulator to compare the new interface to the current 777 interface (Prada, Mumaw, Boehm-Davis, & Boorman, 2007). Results from these experiments can be compared with model predictions to show the usefulness of the list modeling approach.

Boeing Pilot Performance

Boeing gathered performance data on flight tasks in a medium-fidelity, setting to get feedback on proposed interface improvements and to generate performance data comparing the 777 design to the proposed design (Prada et al., 2007). Two desktop computer simulations of the 777 and proposed automatic flight control panels and associated displays were created. The simulations provided appropriate feedback, including mode changes, as controls were manipulated. However, the aircraft remained frozen in time and space until advanced by the experimenter. Participants controlled the simulation using a standard two-button mouse. For this paper, only data from the 777 interface is considered.

Participants The participants consisted of twelve FMC-naïve subjects who were male Boeing employees. All were general aviation pilots with instrument rating. Six had commercial certification and four were not instrument current. They had no previous exposure to the 777 FMC.

Procedure Twenty training tasks were selected to capture tasks that are difficult on each interface and to provide a representative set of functions. In the training tasks, for each action (click) on the interface, the location and time were collected. Also collected were overall task time, number of steps correct, and trials to mastery.

Results The number of steps in the tasks ranged from two steps to thirteen steps. For this paper, tasks are grouped into those with an average of two, four, seven, and thirteen steps. Trials to mastery increased with the number of steps in the task (Figure 4).

Model Performance

The original list model of Anderson et al. (1998) made predictions for lists with three items up to twelve items. The current model retains this range, and so, for analysis, tasks with two steps are compared to lists with three items and tasks with thirteen steps are compared to lists with twelve items (four steps are compared directly, as are seven).

Results Model runs with the step time of 5 seconds used by Matessa and Polson (2006) show trials to mastery increasing with the number of steps in the task. The difference in trials to mastery between the model and subjects averaged 1.5 trials (Figure 4, model-pre).

A post-hoc analysis used the actual average step time from subjects as input to the model. For tasks with an average of two, four, seven, and thirteen steps, the average step time was 15.2, 8.1, 8.0, and 6.5 seconds, respectively. The difference in trials to mastery between this model run and subjects averaged 0.8 trials (Figure 4, model-post).

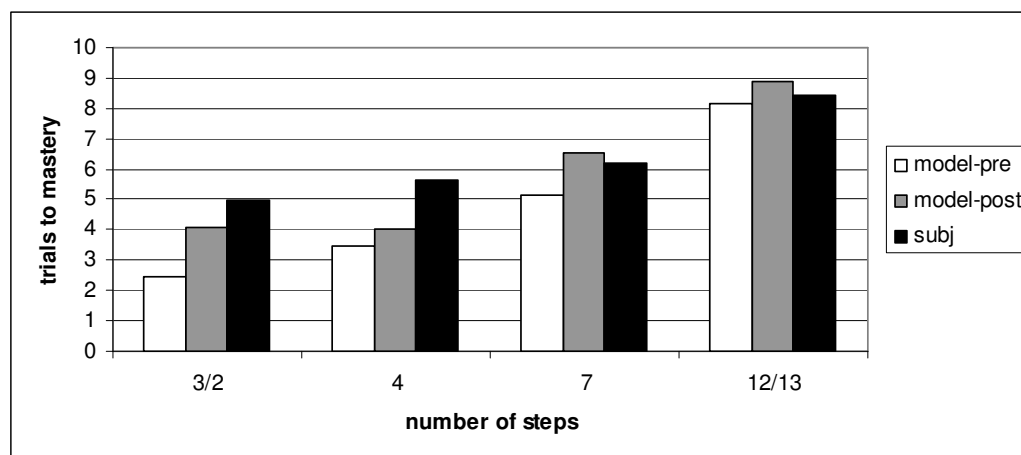


Figure 4: Trials to mastery for model and subjects.

Conclusions

The benefit of the list model representation for making training predictions can be seen in the accurate *a priori* predictions of trials to mastery given the number of task steps. The benefit of using accurate step times can be seen in the even more accurate post-hoc model results.

Ideally, the list model would be given an accurate estimate of step times without seeing the data ahead of time. To this end, the list model is currently being integrated with CogTool (John, Prevas, Salvucci, & Koedinger, 2004). CogTool takes as input a demonstration of an interface task and returns a zero-parameter prediction of task performance time based on ACT-R primitives. With this information, the number of steps in the task and average step time can be fed into the list model in order to make training predictions. A number of open issues remain, such as the level of abstraction of a “step”. Does a step to push a button include the visual search for that button, or is that a separate step? More empirical work is needed to determine in what situations the list model representation can be useful in training prediction.

Acknowledgments

Funding for this work was provided by the National Aeronautics and Space Administration.

References

- Air Transport Association. (1999). *Performance of Standard Navigation Tasks by FMS-Generation Aircraft* (Third report by the Human Factors Committee Automation Subcommittee).
- Anderson, J. R. (2007) *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.
- Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology*, 5, 451-474.
- Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An Integrated Theory of List Memory. *Journal of Memory and Language*, vol. 38, 1998, pp. 341–380.
- BASI (1999) Advanced Technology Safety Survey Report. Flight Safety Digest Special Issue. Flight Safety Foundation, June-Aug 1999, pages 137-216.
- Catrambone, R. (1995) Aiding subgoal learning: Effects on transfer. *Journal of Educational Psychology*, 87, 5-17.
- Ebbinghaus, H. (1885/1913). *Memory: A contribution to experimental psychology* (Henry A. Ruger & Clara E. Bussenius, Trans.). New York: Teachers College, Columbia University.
- Federal Aviation Administration (FAA) Human Factors Team. 1996. Report on the Interfaces between Flightcrews and Modern Flight Deck Systems (June 18, 1996). Washington: U.S. Department of Transportation, Federal Aviation Administration.
- John, B., Prevas, K., Salvucci, D., & Koedinger, K. (2004) Predictive Human Performance Modeling Made Easy. *Proceedings of CHI, 2004* (Vienna, Austria, April 24-29, 2004) ACM, New York.
- Matessa, M., & Polson, P. (2006). List Models of Procedure Learning. *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics* (pp. 116-121), San Francisco, CA.
- Pavlik, P. I., Jr., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29, 559-586.
- Polson, P. G., Irving, S., & Irving, J. E. (1994). Final report: Applications of formal methods of human computer interaction to training and the use of the control and display unit. Washington, DC: System Technology Division, ARD 200, Department of Transportation, FAA.
- Prada, L. Ricardo; Mumaw, R J.; Boehm-Davis, D. A.; Boorman, D.J. (2007) Testing Boeing's Flight Deck of the Future: A Comparison Between Current and Prototype Autoflight Panels. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, Aerospace Systems, pp. 55-58(4).
- Salas, E., Bowers, C.A., and Prince, C. eds. (1998). Special Issue: Simulation and Training in Aviation. *International Journal of Aviation Psychology*, 8(3).
- Salas, E., Bowers, C. A., & Rhodenizer, L. (1998). It is not how much you have but how you use it: Toward a rational use of simulation to support aviation training. *International Journal of Aviation Psychology*, 8(3), 197-208.
- Taatgen, N.A., Huss, D. & Anderson, J.R. (2008). The Acquisition of Robust and Flexible Cognitive Skills. *Journal of Experimental Psychology: General*, 137(3), 548-565.