Instructional Representation in Cognitive Modeling

David Huss (dhuss@andrew.cmu.edu) Niels A. Taatgen (taatgen@cmu.edu), and John R. Anderson (ja@cmu.edu)
Carnegie Mellon University, Department of Psychology, 5000 Forbes Av.
Pittsburgh PA, 15213, USA

Our previous research demonstrated how a cognitive architecture can inform the design of instructions. This abstract describes continuing research into instructional representations in both human instruction and cognitive architectures.

In our past study (Taatgen et. al, submitted), two different forms of instructions were devised. These were dubbed "list-style" and "operator-style." As the name list-style implies, procedural information is represented as a list of steps. These are common in cognitive models and human instruction. The list-style instructions for this experiment were taken directly from a major airlines training manual.

The operator-style instructions take the basic format of "if you are in this state: then do this: and the result will be this" (if/then/result). Further, the "if" part of the operator was assumed to follow Taatgen's (in press) minimal control principle and rely as much as is practical on external state information. See Taatgen, Huss & Anderson (submitted) for more elaboration on these instruction styles.

Predicted benefits of operator style instructions were their generalizability and error recovery. We tested this, by providing our subjects with trials (referred to as "hard" trials) that required extrapolation from their initial instruction. In other words, if the subjects simply followed the instructions as they were given, they would not successfully complete these tasks.

Also predicted was that the operator condition would better handle trials that were partially completed. This is based on the claim that operator instructions are biased towards reliance on external state information and should better be able to pick up a trial that is mid-task. This claim was not previously tested and forms the basis for this experiment.

Experiment

Our experiment is slight modification to a previous experiment (Taatgen, Huss & Anderson, 2006). As such, only a brief overview will be given.

The Flight Management System (FMS)

As in our previous study, our experiment is constructed around a simulated FMS. The FMS is a complex computerized interface responsible for the automated navigation in most modern aircraft. The FMS interface is notoriously difficult to learn and can even challenge experienced pilots (Sherry, Polson, Fennell & Feary, 2002). All of the tasks in this experiment involved the LEGS page functionality of the device.

Trial types

6 different types of trials were used in this experiment:
1. *Easy Direct-to's*-Fully trained, present in old experiment
2. *Easy Discontinuities*-Fully trained, discontinuity requires using a sub-procedure, present in old experiment
3&4. *Hard Direct-to's and Discontinuities*–Require extrapolation, present in old experiment
5&6. *Copilot and Wrong Copilot*-Partially completed problems encompassing types 1-4 (either done correct or incorrectly), new to this experiment

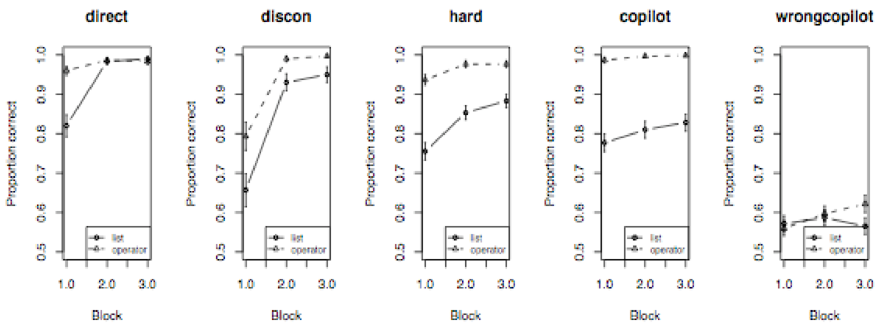
Participants

40 students and adults from Carnegie Mellon University and the surrounding community were paid volunteers in the experiment (20 in the list-style condition and 20 in the operator-style condition).

Procedure

As in our initial study subjects were first given a general overview of the FMS interface and basic training. Following this, they were given specialized instructions on the completion of an FMS procedure in either list or operator-style formatting.

The main part of the experiment consisted of 3 blocks of 24 trials (3 each of trial types 1-4). The first 12 trials of each



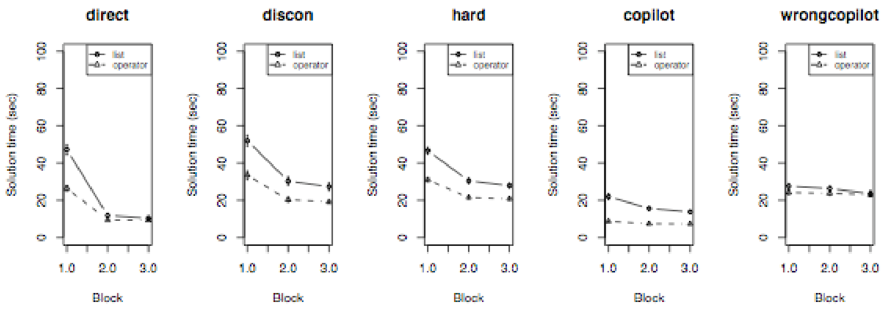
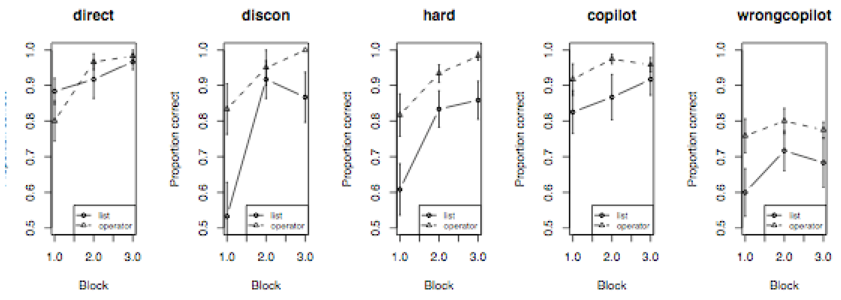


Figure 1: Model predicted performance
Figure 2: Model predicted latency



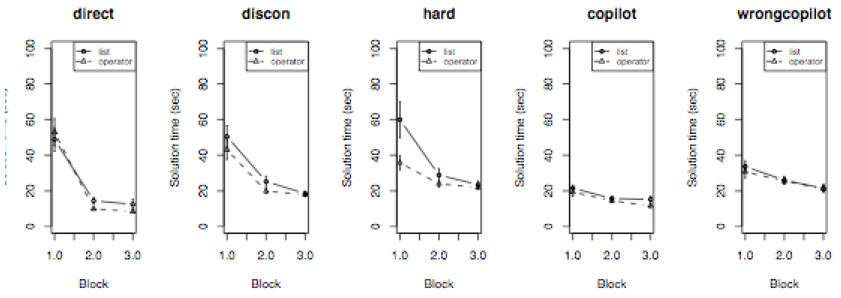


Figure 3: Human performance data
Figure 4: Human latency data

block replicated the trials in our previous experiment. The final 12 consisted of random partially completed problems.

The Model

We applied our existing ACT-R (Anderson et. al, 2004) model to this new experiment. The model was not modified nor were any of the parameters re-fit. Despite these limitations it replicated our previous results and handled the partially completed problems.

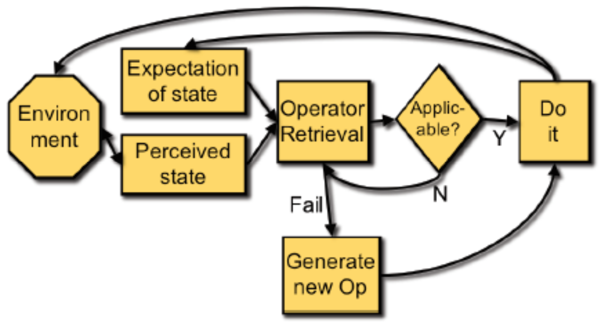


Figure 5: Overview of the model

The model's ability to handle the new trials is significant. A frequent criticism of cognitive models is their brittleness. When faced with an unexpected state, many models fail.

The model owes its success to its method of operation (figure 5). All the procedural steps are stored in the same format (if/then/result). In the list-condition, the if/then refer to the previous and next steps (step1/do step2/step3. In the event an operator is forgotten, the model tries pushing a random key until something works, stores what does, and thus forms new operators. The result is an ability to handle novel situations and a gradual formation of operator-style operators even in the list condition. See Taatgen, Huss & Anderson for a more detailed explanation (submitted).

The model predicts that the operator-style instructions should be slightly faster (figure 2). It also predicts that, while performance will be similar in the direct-to condition (the task that was explicitly trained on), the operator-condition is should result in superior performance in all other trial types (figure 1).

Results

Figures 3 and 4 show the accuracies and latencies obtained in our human data. Dashed lines represent the operator-style condition while solid lines are the list-style instructions. Despite considerable variation, as evidenced by substantial error bars, significant effects exist.

As predicted, performance on the direct condition tasks is comparable. Also, the operator-style condition performs better on both the harder problems and those involving a discontinuity. The difference in performance in the correct copilot problems is considerably less than expected. The overall performance for the wrong-copilot problems was higher and with a greater than predicted difference between the two conditions. Our previous results are replicated in our new data (Taatgen et. al, submitted).

Conclusion

Taken in context with our previous study, this experiment provides further support for the conclusion that instructions designed on the basis of a cognitive model can indeed produce significant improvements in performance. Through the inclusion of partially completed problems, we have demonstrated the robust nature inherent in operator-style instructions.

Finally, we present an example of how cognitive models can effectively handle error conditions and novel states. The concepts of operator retrieval, testing, and generation are applicable to most ACT-R models.

Acknowledgments

This research is supported by NASA grant NRA2-38169. We would like to thank Peter Polson and Karl Fennell for their advice.

References

Anderson, J.R., Bothell, D., Byrne, M., Douglass, D., Lebiere, C. & Qin, Y. (2004). An integrated theory of mind. *Psychological Review, 111* (4), 1036-1060.

Sherry, L., Polson, P., Fennell, K., Feary, M. (2002). Drinking from the Fire Hose: Why the Fmc/Mcdu Can Be Hard to Learn and Difficult to Use. Honeywell internal report C69-5370-022.

Taatgen, N. (in press). The Minimal Control Principle. In W. Gray (Ed.), *Integrated Models of Cognitive Systems*. Oxford University Press.

Taatgen, N., Huss, D., & Anderson J.R., (submitted). The Minimal Control Principle, *Proceedings of the 2006 International Conference on Cognitive Modeling.*