SAND REPORT

SAND2002-3727
Unlimited Release
Printed December 2002

A Review of Production System Models of Cognition and Example Demonstration

Nathan G. Brannon.

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2002-3727
Unlimited Release
Printed December 2002

A Review of Production System Models of Cognition and Example Demonstration

Nathan G. Brannon, Ph.D.
Reliability Assessment & Human Factors
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-0830

Abstract

There have been significant efforts to develop cognitively plausible software architectures of human information processing in the last three decades. This report summarizes several architectures that continue to be developed. The specific type of cognitive models developed are known as production system architectures, which refers to the characterization of knowledge in terms of procedural ("how-to" knowledge) condition-action relationships consisting of declarative ("what" or factual) knowledge. To illustrate the ability for these models to instantiate human cognitive performance, a simulation using ACT-R (Adaptive Control of Thought – Rational) was implemented for a supervisory control task. Correlations between simulated and human learning of the task were measured and yielded correlations as high as 0.93.

1. Introduction and Background

A relatively small but active area of research exists in the cognitive science arena that is often referred to as production system or rule-based modeling (Laird, Newell, and Rosenbloom, 1987; Just and Carpenter, 1992; Kieras and Meyer, 1997; Anderson and Lebiere, 1998). This report summarizesf production system models and reviews the effectiveness of a production system that models a supervisory control task. Model summaries include some comments concerning their ability to account for knowledge degradation (Brannon, 2001; Brannon and Koubek, 2001) or, in other words, the ability to account for mechanisms that could lead to forgetting or a reduced ability to retrieve knowledge from memory.

One early hurdle in understanding production systems is simply becoming familiar with the unconventional terms associated with this area of research. A logical starting point is with the term "production." Production comes from a fundamental assumption that human cognitive behavior is goal oriented and can be characterized in terms of condition/action relationships. Such If-Then rules are known as "productions" or "production rules." A "production system" is an organized collection of If-Then rules representative of human information processing.

Knowledge has been characterized in a variety of ways in psychology. Production systems categorize knowledge in two dimensions, declarative and procedural. Declarative knowledge represents facts (the "what" knowledge) such as phone numbers or state capitals. Procedural knowledge is the "how to" knowledge integrating declarative knowledge into productions. Examples of procedural knowledge include mathematics or monitoring a nuclear power plant.

The most common motive behind the development of production system models is for the validation of cognitive theories. The application of production systems as subsystems in the design process is rare and only beginning to be explored (Brannon, 2001).

2. Production System Modeling of Cognition

Several production system architectures will be reviewed and are summarized in Table 1. While many dimensions could be depicted in the rows, the sample topics were chosen with an emphasis on knowledge dynamics.

Before going into detail concerning production systems, it should be noted that several models of cognition exist other than production system architectures. Examples include

Meta Trouble Shooter (Meta-TS; Ram, Narayanan, and Cox, 1995), and the Operator Function Model (OFM; Mitchell, 1987). Tools such as the OFM and Meta-TS place more emphasis on actions than atomic levels of cognition. Mitchell, Rubin, and Govindaraj (1986), for example, notes that the OFM is not intended to model the inner workings of the mind. In contrast, production systems begin with a representation of cognitive processes and extrapolate the mechanisms to external tasks.

Table 1. Production System Architectures of Cognition

	3CAPS	EPIC	SOAR	ACT-R
Purpose	Problem solving and language comprehension	Simulate multi-task performance and human information processing	Problem solving and learning	Problem solving and learning
Theoretical Underpinnings and Preceding Framework	Individual differences and capacity theory (Just and Carpenter, 1992)	GOMS and the Model Human Processor (Card, Moran, and Newell, 1983)	Problem space and operators (Newell and Simon, 1972) and OPS (Forgy, 1995)	ACT (Anderson, 1976) and ACT* (Anderson, 1982)
Learning Mechanisms	Production firing in connectionist networks	None	Chunking	Production compilation and subsymbolic learning
Degradation Mechanisms	Displacement	None	None	Decay and competition
Knowledge Metrics	Level of activation	Reaction time	Chunks	Expected gain and strength
Validation	Simulation vs. human data for reading tasks (Haarmann, Just, and Carpenter, 1997)	Simulation vs. human data measuring reaction time (Kieras, Wood, and Meyer, 1997)	Varied (Rosenbloom, Laird, and Newell, 1993)	Varied, particularly in mathematics education (Anderson and Lebiere, 1998) and programming (Anderson, Farrell, and Sauers, 1984)

2.1. 3CAPS

Just and Carpenter (1992) developed the Concurrent, Capacity-constrained, Activation-based, Production System (3CAPS) for modeling language comprehension and problem solving. A central feature of the model is the representation of individual differences in working memory capacity. The greater the working memory capacity is, the greater is the ability to comprehend language. Working memory in 3CAPS is organized in a connectionist network. Knowledge is measured by the level of activation, and production firings control the flow of activation. Typical means of validating the model include reading tasks where human data are compared with 3CAPS data (Just and Carpenter, 1992; Haarmann, Just, and Carpenter, 1997).

With regard to knowledge degradation mechanisms, 3CAPS uses a concept known as displacement. Displacement means that if the level of activation exceeds working memory capacity, knowledge is lost, and information-processing speeds decrease as well.

The older the knowledge, the more likely it will be forgotten should the level of activation exceed working memory capacity.

The use of displacement as a general mechanism of knowledge degradation has been criticized. Nairne (1996: p. 77) argues that since working memory is not necessarily fixed, the displacement mechanism has "fallen out of favor" with researchers. However, if the working memory capacity of an individual is accounted for (as performed by Just and Carpenter, 1992), then displacement is a viable mechanism of interference.

Although 3CAPS is a production system model, the emphasis upon language comprehension and individual differences in working memory capacity is not within the scope of the current research. This study intends to add insight to factors in interface design rather than the arrangement of lexical or syntactic information. Other models are more generalized accounts of human performance, which is better suited for this study.

2.2. EPIC

A tool developed for modeling human information processing is GOMS (Goals, Operators, Methods, and Selection Rules; Card, Moran, and Newell, 1983). GOMS provides a means of describing human performance through a sequence of operators (Eberts, 1994). Operators are cognitive, perceptual, or motor acts that change the user's mental state (Card, Moran, and Newell, 1983). EPIC (Executive Process Interactive Control; Meyer and Kieras, 1997a) is a production-system architecture that was built upon the GOMS framework.

EPIC possesses several useful features. From an applied perspective, an attractive characteristic of EPIC is the computational modeling of multiple-task performance (Meyer and Kieras, 1997b). EPIC took an early lead in the effort to account for the relationship of sensory/perceptual processing and cognitive processing. Perceptual processors included in EPIC include visual sensory, visual perceptual, auditory perceptual, and tactile. EPIC also accounts for motor processing with manual motor, ocular motor, and vocal motor processors. EPIC primarily measures knowledge using factors such as reaction time.

Unlike other production system models, EPIC does not provide mechanisms for learning and degradation. However, other production systems such as ACT-R have been augmented with EPIC-like perceptual motor characteristics, and one version is known as ACT-R/PM (perceptual motor; Byrne, 2000). ACT-R/PM unites the human information processing strengths of EPIC with the learning mechanisms of ACT-R.

2.3. SOAR

From their inception, tools such as ACT-R (Anderson, 1982; Anderson, 1998) and SOAR (State, Operator, and Result; Laird, Newell, and Rosenbloom, 1987; Newell, 1990) have included mechanisms for knowledge acquisition. SOAR possesses a refined structure enabling knowledge acquisition and efficient information processing. All tasks are formulated in what is referred to as a "problem space" (Newell and Simon, 1972). Elements of decision making, for example, can include current states and means by which a desired state is reached. Goals are strictly generated through an automatic subgoaling mechanism. The subgoaling mechanism is a fundamental component of SOAR's learning capability. SOAR stores long-term memory simply as productions, rather than distinguishing declarative and procedural knowledge. Long-term memory supports problem solving. For example, in decision making, a production can be recalled from long-term memory that helps to quickly make a choice among alternatives.

SOAR primarily acquires knowledge through chunking. The learning process begins when SOAR is provided a task and chooses a problem space. Within the problem space, active operators can be chosen to change the current state to a desired state. If a unique activation cannot be derived, SOAR has reached an impasse. Subgoals are then automatically generated to resolve the impasse. If the impasse is still not resolved, more subgoals are generated until the impasse is resolved. The results are permanently cached as productions (analogous to "chunking" in human cognition). These productions store the means by which the impasse was resolved. Therefore, when the same goal is set in the future, the stored production fires without the need for generating subgoals. This results in more efficient performance.

The refined structure of SOAR has enabled extensive applications for design and validation (Rosenbloom, Laird, and Newell, 1993; Pew and Mavor, 1998). Although the simplified nature is advantageous from an applied standpoint, this limits the scope of cognitive variables that can be computationally modeled (Anderson, 1993). Newell (1990: p. 309) has conceded this point, stating that "the assertion that chunking is a sufficient mechanism should be considered a speculative and a priori unlikely hypothesis."

Koubek et al. (1999) details multiple variables associated with learning. To model cognition adequately, computational models must explore multiple mechanisms of learning. The most active production system that has made progress in this respect is ACT-R.

2.4. ACT-R

ACT-R (Adaptive Control of Thought – Rational; Anderson, 1993; Anderson and Lebiere 1998) has a rich history beginning with an early architecture known as HAM (Human Associative Memory; Anderson and Bower, 1973). By 1976, Anderson had expanded the focus of his research from memory to include learning in establishing ACT. Meanwhile computational models accompanied the theoretical underpinnings to simulate cognition. The computational models included versions from ACTA to ACTE. Further refinement led to the development of ACT* (Anderson, 1982). Although the learning mechanism Anderson refers to as "analogy" is mentioned, the mechanism did not really fit into ACT*. Furthermore, the learning mechanisms that were included in ACT* were difficult to support empirically. As a result of these limitations, ACT-R 2.0 was developed (Anderson, 1993). Further enhancements including more powerful declarative representations and more stable mechanisms for procedural knowledge were added to ACT-R 4.0 (Anderson and Lebiere, 1998).

The basic structure of a production in ACT-R consists of three stages: the goal condition, chunk retrieval, and goal transformation. The ACT-R process is summarized in Figure 1. The three stages are depicted with their respective elements. Goals are organized in a stack. The term "stack" is appropriate because of the first in, last out (FILO) organization of goals in ACT-R. The process initiates with a goal being pushed (lower left-hand corner; "Start Cycle with Goal"). This tells ACT-R to focus its attention on a specific goal. Once a goal is chosen, productions are derived from the general knowledge base that match the goal. If there are no productions that match the goal, the goal is "popped" with failure. To pop a goal is to remove it from the focus of ACT-R's attention. As a result of "failure," ACT-R cannot currently achieve the goal, and this typically results in the return to the higher-level goal that set it.

Normally at least one production is available, and if so, productions are ranked based on their expected gain (E). The following is the formula for expected gain:

$$E = PG - C$$

In a sense, this is the difference of benefit (PG) and cost (C). P is the probability that the goal will be achieved if that production rule is chosen. P is derived from parameters associated with the probability of successful execution of the production and the likelihood the related goal will be achieved should the production be executed successfully.

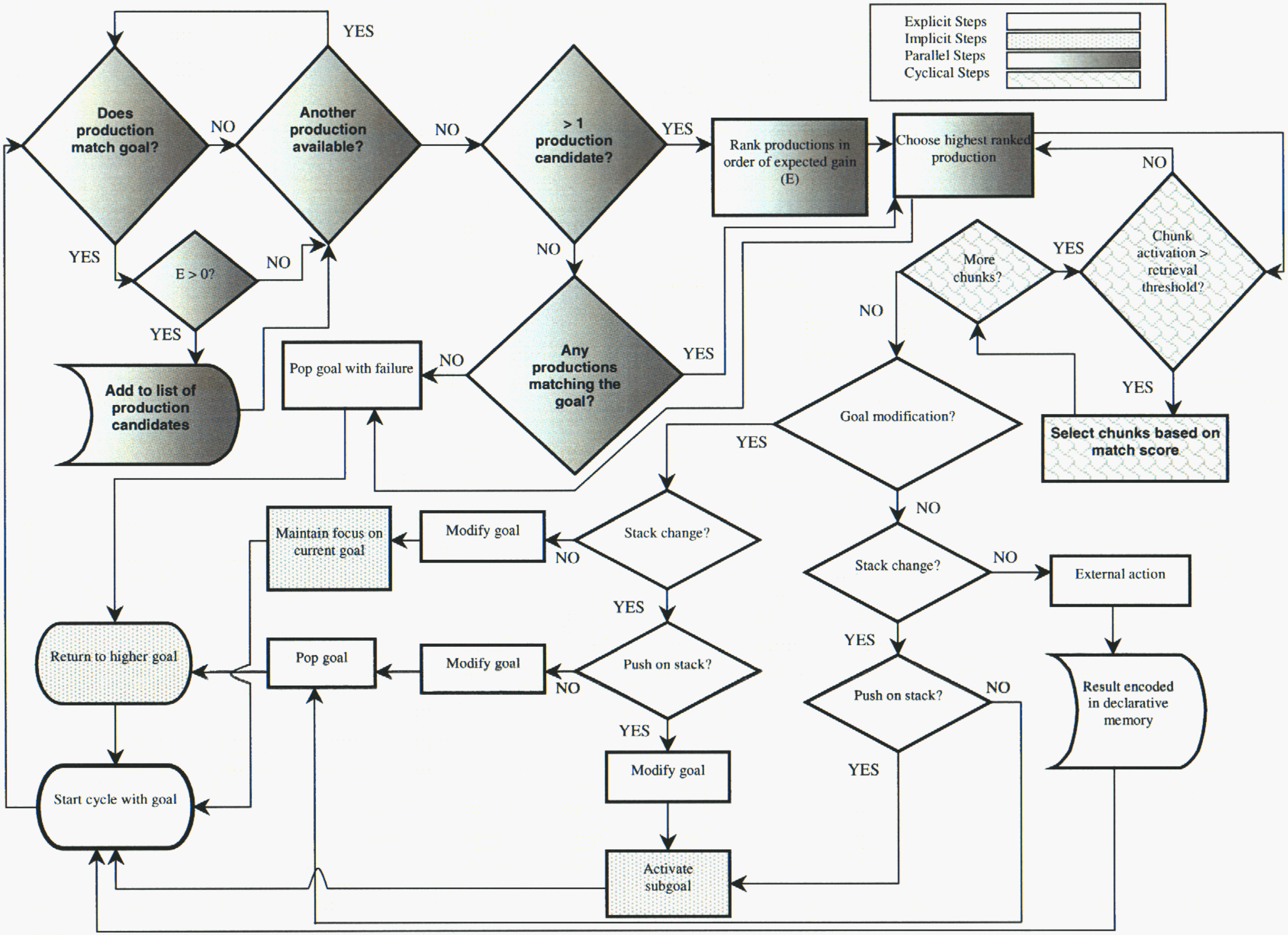


Figure 1. ACT-R 4.0 Production Cycle

The parameter G is defined as the value of the goal. The calculation of G is subjective and is related to the time ACT-R should be interested in achieving the goal. Anderson and Lebiere (1998: p. 63) admit ACT-R has "little to say" with regards to the initial value of G. However, once experience accrues, G can play a greater role in the expected gain formula. For example, one of the outcomes of a production is the generation of subgoals. It is possible that deeper and deeper levels of subgoals are created, and as these levels grow deeper, the goal becomes associated with cumbersome requirements to achieve the goal. Consequently, the parameter G decreases, thereby decreasing the overall expected gain (E) of the top goal. Efforts to quantify supporting parameters exist but are in early stages (Belavkin and Ritter, 2000).

When a particular production is chosen, there is an associated cost (C) of achieving the goal. Cost is derived from factors such as the expected effort. The time needed to retrieve declarative knowledge and actual performance times are used to determine the expected effort. Other subsymbolic variables used to derive cost include prior successes or failures.

Returning to Figure 1, the expected gain helps to determine the optimum production that matches the goal. However, the highest ranked production is not necessarily the production that will be executed. If expected gain is less than zero (i.e., cost is greater than the benefit), the goal will pop with failure. Another context in which the highest ranked production will not be chosen is when there are difficulties associated with chunk retrieval.

Productions consist of chunks and the likelihood of retrieving any chunk is determined, in part, by that chunk's level of activation. A parameter fundamental to the calculation of activation is base-level activation. The recency and frequency with which a chunk is used determines the base level of activation. Strength of association between a chunk and possible elements (e.g., other chunks or productions) as well as attentional weighting are accounted for in the level of activation. Attentional weighting has been examined in relation to working memory capacity (Lovett, Reder, and Lebiere, 1999).

If enough time passes, decay can decrease the level of activation to a point where the chunk can no longer be retrieved. ACT-R accounts for this with a threshold for level of activation. The probability of exceeding this threshold uses parameters including the level of activation, decay rate (typically set at 0.5), and a noise-control parameter. The noise parameter is derived from logistic distributions.

Most chunks do not perfectly match a production, and without an account for the degree with which the chunk matches and mismatches the production, there is a risk of commission error for chunks above the retrieval threshold. ACT-R provides what is called a "match score." This score is simply the difference between the level of activation and the degree of mismatch. A chunk pattern, in the condition side of a production, contains slots, and the number of slots in which a chunk mismatches the desired chunk pattern determines the degree of mismatch.

Once a chunk is retrieved, the production can be executed. Execution can result in one of the six following outcomes:

1. **No goal modification. No change in the goal stack.** The purpose of this type of production is to generate an external action. As described by Anderson and Lebiere (1998), there has been active development of ACT-R/PM (Perceptual Motor; Byrne, 2000) to account for interactions with an external environment. This version of ACT-R contains components and mechanisms similar to EPIC (Kieras, Wood, and Meyer, 1997).

2. **Goal modification. No change in the goal stack.** In this case, the goal will be modified, but the current goal will remain the focus of attention upon completion of production firing.

3. **No goal modification. Push on stack (subgoal initiated).** Often subgoals are needed to solve more specific problems. This type of production initiates a subgoal, which is similar to subgoaling in SOAR (Rosenbloom, Laird, and Newell, 1993). In ACT-R, once the subgoal is executed, the focus returns to the higher goal that called it without the goals being modified. An interesting difference worth noting between SOAR (Rosenbloom et al., 1993) and ACT-R is that SOAR stores the result of subgoals as productions whereas ACT-R stores results of subgoals in the form of chunks.

4. **Goal modification. Push on stack (subgoal initiated).** A goal can be modified before a subgoal is initiated. When the subgoal is completed, it is sometimes useful to have a different goal to return to so that the same production rule is not fired repeatedly.

5. **No goal modification. Pop stack.** It is useful to terminate a recursive loop of productions as in the case of adding columns in mathematics. There could be a

production called Do-Add-Stop that, once the condition is met, terminates the process by a pop of the goal stack.

6. **Goal modification. Pop stack.** Often, in simple productions such as the addition of two numbers, this is a useful type of production. The slot for sum can be modified to the solution before the production pops and the focus returns to a higher goal.

It is helpful to note that declarative memory receives the details of the outcomes (such as successes or failures). This is essential to the process of learning and more specifically production compilation.

ACT-R contains two levels of processing and learning: symbolic and subsymbolic. With respect to learning, the symbolic level involves the acquisition of chunks and production rules. The subsymbolic level of learning involves the acquisition of parameters that govern the deployment of elements such as chunks and productions. The subsymbolic level also provides stochastic noise parameters.

Production compilation, where new rules are generated, occurs at the symbolic level of ACT-R. Anderson and Lebiere (1998) note that production compilation was one of the last concepts to be included in their work. Furthermore, it is acknowledged that validation studies are still being conducted, and therefore, the production compilation mechanism of learning remains "somewhat tentative" (p. 117).

Currently, there are several modifications in the latest release of ACT-R (5.0; Lebiere, 2001). Additional standard features will include perceptual motor buffers derived from ACT-R/PM. The processing of information in the buffers will be parallel, while the cognitive elements of ACT-R will remain serial. The buffers will test conditions in the left-hand side (or the condition) of productions, and buffer actions will be available in the right-hand side as well. Buffer types include visual, manual, aural, vocal, goal, and action.

Another significant development in version 5.0 is the refined nature of production compilation. Collapsing consecutive productions together can generate new productions. The resulting production is a specialized version of the parent productions.

3. Demonstration

A significant objective of this report is also to illustrate a computational representation of human performance on a resource management task. The task was chosen to demonstrate the ability to model a more real-world task relative to tasks typically modeled by tools like ACT-R. As a result, the process by which procedural knowledge degrades can be better understood rather than treating the process as a "black box" with only input/output relationships. The computational framework chosen for this component of the research was ACT-R (v. 4.0) (Anderson and Lebiere, 1998). With early research dating back as far as Anderson and Bower (1972) and ever-expanding applications, ACT-R is arguably the most empirically rich and active computational model of cognition.

3.1. Task and Model Formulation

A vehicle for chosen for demonstrating a production-system model was the Multi-Attribute Task Battery (MATB; Comstock and Arnegard, 1992). MATB (Figure 2) provides a suite of tasks analogous to real-world fields such as air-traffic control and waste-water management. The specific task in MATB used for this research was the Resource Management task. Note that during data collection involving human participants, all the other tasks (e.g., communication) were covered using a piece of cardboard, exposing only the resource management section of the MATB interface.

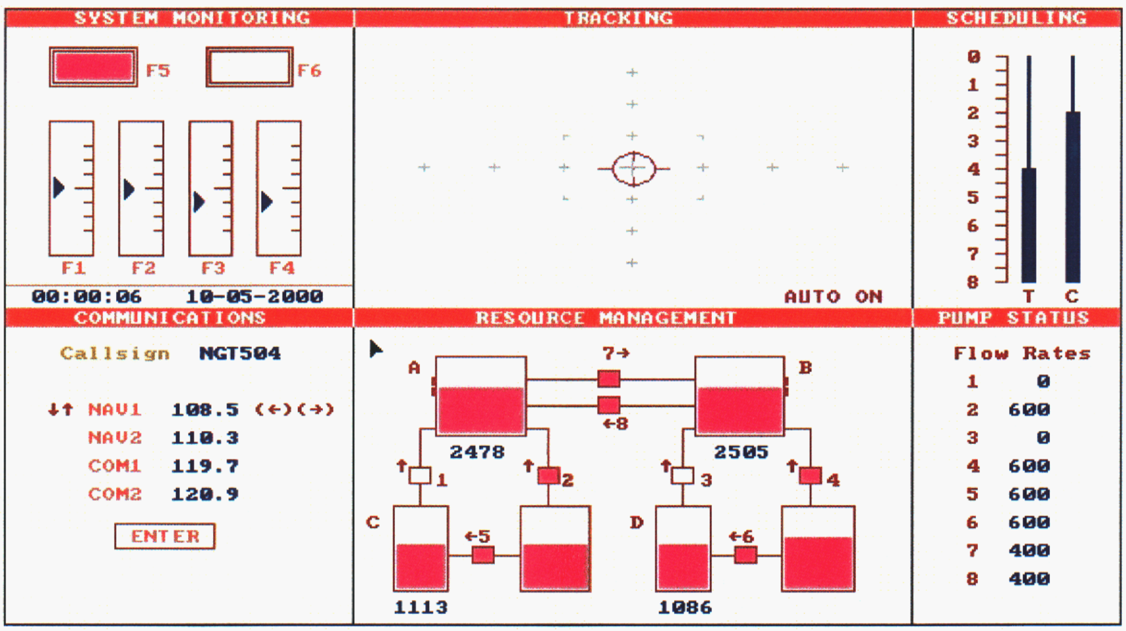


Figure 2. MATB Visual Interface

The goal of the Resource Management task is to maintain fluid levels in tanks at pre-specified levels. Participants toggle pumps ON/OFF to control flow in and out of the tanks. Pumps are labeled by numbers, which map to a corresponding key on the keyboard. For example, to turn on pump 1, subjects can press 1 on the keyboard. If the key is pressed again, the pump is turned off.

Since different pumps provide varying flow rates and certain tanks have certain target levels, participants must learn a pattern of actions to minimize the variability around the target level. An added source of complexity is that tanks A and B lose resources at a rate of 800 units/minute. Although participants are informed of this factor, the information is not explicitly displayed on the interface. As participants' procedural knowledge becomes refined, root mean square error (RMSE) was expected to decrease based on pilot studies.

MATB allows quantitative information about performance to be easily collected. The software was modified so that every time a key was pressed (e.g., to turn a pump ON/OFF), a time stamp [time (seconds) into the trial], along with the current tank levels would be recorded in a separate data file. Additional modifications provided a redundant time stamp and tank-level recording every second. Such recording features allow calculations of RMSE and user input rate (inputs/second).

MATB is written in QuickBASIC (v. 4.5) and is less than 1 megabyte in size. MATB runs from a DOS prompt and is capable of being used on a 386 PC or higher. Output files are tab-delimited for easy transition to statistical analysis packages.

With respect to performance measures, the primary measure was RMSE. Although participants are instructed to maintain a target tank level, pilot studies found that the actual mean would often deviate from the target level (typically higher). RMSE accounts for such a control bias unlike standard deviation. Therefore, RMSE is a more comprehensive measure of variability and performance. RMSE was calculated cumulatively (for each trial) and in segments (every 50 seconds in a 20-minute trial).

3.2. Control Group Correlations

With the disparity in results for human and ACT-R data found by Brannon (2001), it was thought useful to at least measure how well ACT-R predicted target task and control group trials. Each human control was paired with the ACT-R counterpart to measure their correlation. The correlations were measured with respect to RMSE (Table 2).

Given the fact that there were two observations per treatment combination, each trial for the same treatment combination was averaged. For example, participant 3 had the same experimental condition as participant 32. Therefore, the RMSEs for each trial of participant 3 were averaged with each trial of participant 32. The same approach was used for human and ACT-R data, and the correlation was conducted across trials. This approach avoids pairing and ACT-R and human counterparts individually that could appear to be biased. Individual comparisons were calculated and plotted (Appendix).

Generally, the correlations are high, with the exception of 7 and 20. This effect will be addressed in the following discussion.

Table 2. Control Group Correlations Between Human and ACT-R Data

Participant	*Sample Size*	*Correlation*
3 and 32	5	0.93
4 and 26	7	0.92
5 and 17	7	0.87
7 and 20	9	0.25

ACT-R models predicted human learning with a respectable degree of accuracy. In retrospect, ACT-R was coded with a higher level of noise than needed. Human learning was generally more consistent than expected.

Figures 3 to 6 illustrate RMSE across trials. Although the grouping of participants 7 and 20 reflects a relatively low correlation, Figure 6 illustrates ACT-R adhering to the more typical learning curve, while human performance reflects difficulty in improving performance. Ironically, for participant 5 (see plot in Appendix), the human adhered to a typical learning curve, and ACT-R deviated from the normal pattern. Both human and ACT-R deviations reflect difficulty in gaining task proficiency.

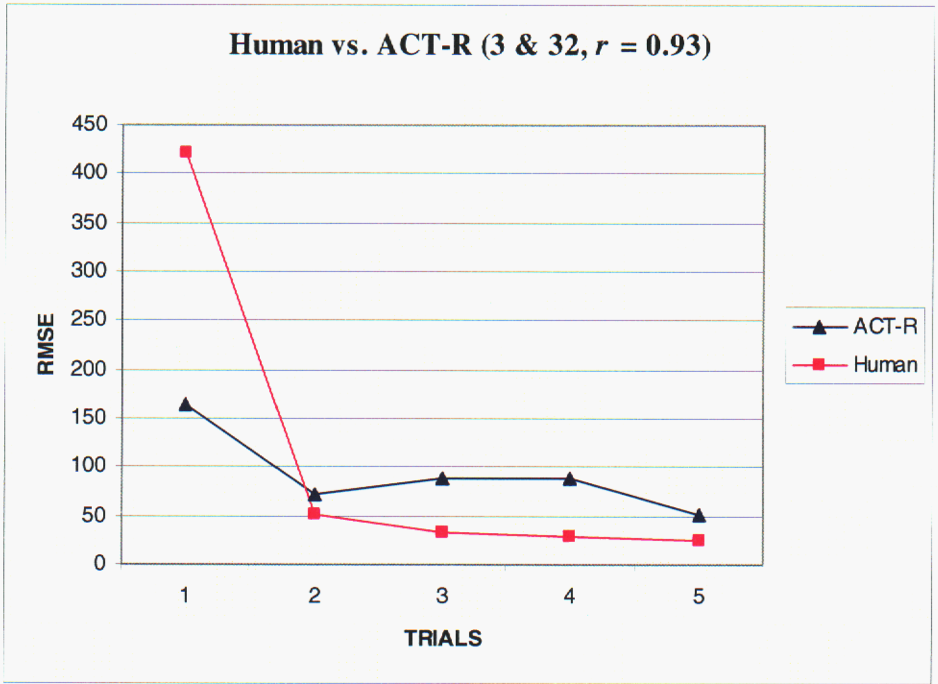


Figure 3. RMSE by Trial for Participants 3 and 32

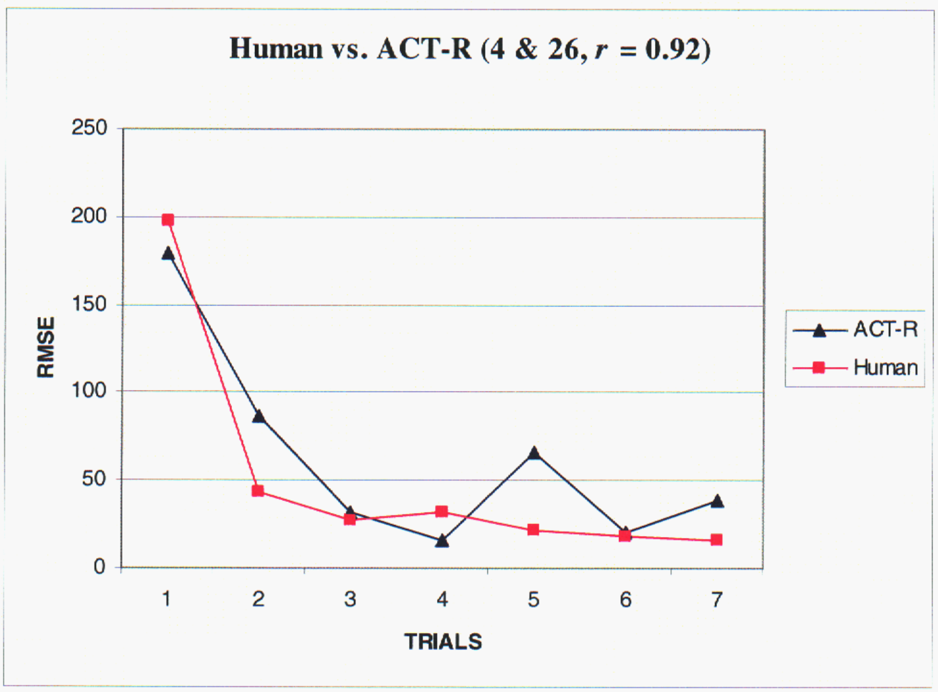


Figure 4. RMSE by Trial for Participants 4 and 26

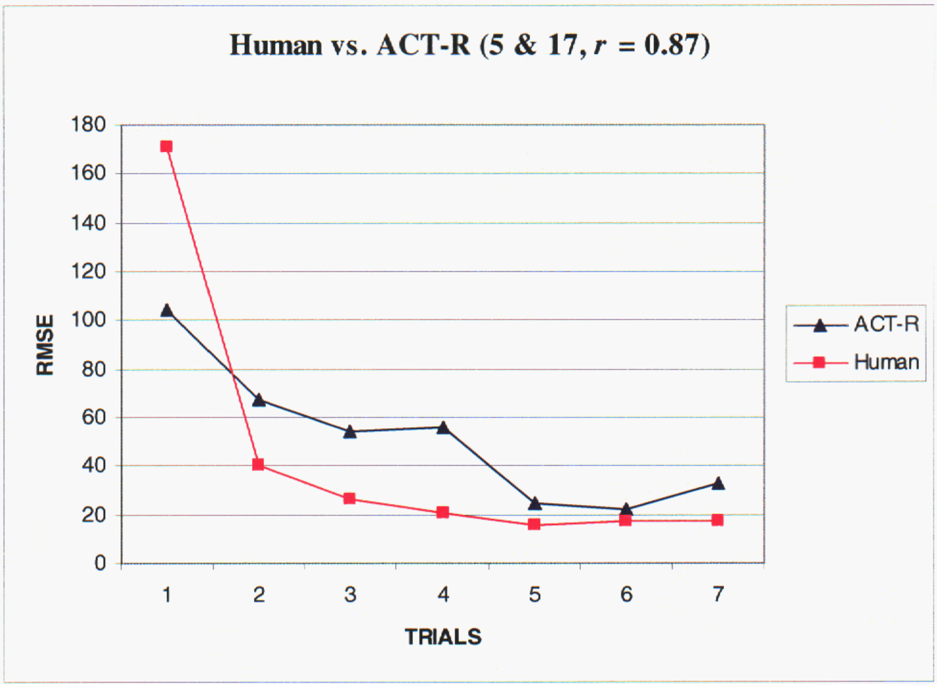


Figure 5. RMSE by Trial for Participants 5 and 17

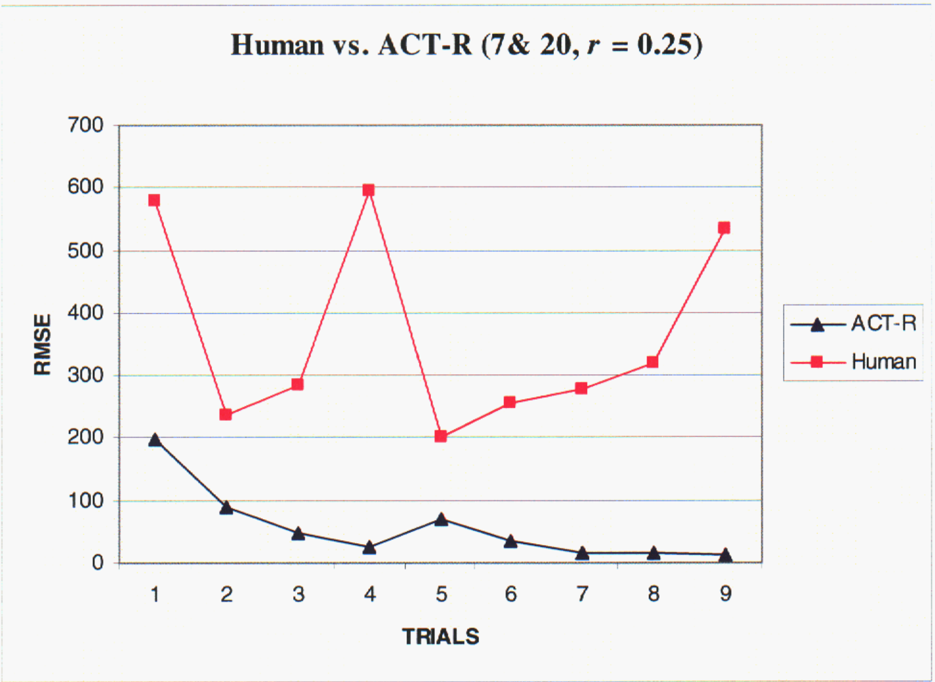


Figure 6. RMSE by Trial for Participants 7 and 20

4. Conclusion

ACT-R successfully performed the resource management task and performed in a manner consistent with literature and the conceptual model. ACT-R remains a promising tool for the simulation and prediction of human cognitive performance. Finally, this simulation was useful in examining cognitive variables in an ecologically analogous supervisory control task.

From a computational standpoint, this research explored new ways to utilize existing parameters within the ACT-R architecture. An even more complex and fertile area of research involves the need to understand effective means of integrating the parameters to further our understanding of how the parameters interact.

Given that most real-world tasks are procedural in terms of knowledge structure rather than declarative, it is essential that research expand its investigation of procedural knowledge dynamics. While some common principles exist between declarative and procedural knowledge, it is proposed that significant differences exist with respect to their susceptibility to interference effects. The added dimensions of performance variability introduced by procedural knowledge are complex.

REFERENCES

Anderson, J. R. 1976, *Language, Memory, and Thought* (Mahwah, NJ: Lawrence Erlbaum Associates).

Anderson, J. R. 1982, Acquisition of cognitive skill, *Psychological Review*, 89, 369-406.

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

Anderson, J. R., Farrell, R., and Sauers, R. 1984, Learning to program in LISP, *Cognitive Science*, 8, 87-129.

Anderson, J. R., and Bower, G. H. 1973, *Human Associative Memory* (Mahwah, NJ: Lawrence Erlbaum Associates).

Anderson, J. R., and Lebiere, C. 1998, *The Atomic Components of Thought* (Mahwah, NJ: Lawrence Erlbaum Associates).

Belavkin, R., and Ritter, F. 2000, Adding a theory of motivation to ACT-R, *7th Annual ACT-R Workshop Proceedings* (Pittsburgh, PA: Carnegie Mellon University, Department of Psychology), August 5–7, 2000.

Brannon, N. G. 2001, *Knowledge Degradation*. Doctoral Dissertation, Wright State University, Dayton, Ohio.

Brannon, N. G., and Koubek, R. J. 2001, Towards a conceptual model of procedural knowledge degradation, *Theoretical Issues in Ergonomics Science*, 2(4), 317-335.

Byrne, M. 2000, RPM 2.0, *7th Annual ACT-R Workshop Proceedings*. (Pittsburgh, PA: Carnegie Mellon University, Department of Psychology), August 5–7, 2000.

Card, S. K., Moran, T. P., and Newell, A. L. 1983, *The Psychology of Human Computer Interaction* (Hillsdale, NJ: Erlbaum).

Comstock, J. R., and Arnegard, R. J. 1992, Multi-attribute task battery for human operator workload and strategic behavior research, NASA Technical Memorandum 104174, NASA Langley Research Center, Hampton, VA.

Eberts, R. E. 1994, *User Interface Design* (Englewood Cliffs, NJ: Prentice Hall).

Forgy, C. L. 1995, OPS languages an historical overview, *PC AI Intelligent Solutions for Desktop Computers,* 9(5), 16-21.

Haarmann, H. J., Just, M. A., and Carpenter, P. A. 1997, Aphasic sentence comprehension as a resource deficit: A computational approach. *Brain and Language*, 59(1), 76-120.

Just, M. A., and Carpenter, P. A. 1992, A capacity theory of comprehension: Individual differences in working memory, *Psychological Review,* 99(1), 122-149.

Kieras, D. E., and Meyer, D. E. 1997, An overview of the EPIC architecture for cognition and performance with application to human-computer interaction, *Human-Computer Interaction,* 12, 391-438.

Kieras, D. E., Wood, S. D., and Meyer, D. E. 1997, Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task, *ACM Transactions on Computer-Human Interaction*, 4(3), 230-275.

Koubek, R. J., Salvendy, G., Tang, E., and Brannon, N. G. 1999, Development of a conceptual model for predicting skills needed in the operation of new technologies, *International Journal of Cognitive Ergonomics*, 3(4), 333-350.

Laird, J. E., Newell, A., and Rosenbloom, P. S. 1987, SOAR: An architecture for general intelligence, *Artificial Intelligence*, 33, 1-64.

Lebiere, C. 2001, ACT-R 5.0 architecture, *2001 ACT-R Post-Graduate Summer School.* (Berkely Springs, West Virginia: Carnegie Mellon University, Department of Psychology), July 21–25, 2001.

Lovett, M. C., Reder, L. M., and Lebiere, C. 1999, Modeling working memory in a unified architecture: An ACT-R perspective, in A. Miyake and P. Shah (eds), *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control* (New York: Cambridge University Press), 135-182.

Meyer, D. E., and Kieras, D. E. 1997a, A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms, *Psychological Review,* 104(1), 3-65.

Meyer, D. E., and Kieras, D. E. 1997b, A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of Psychological Refractory-Period Phenomena, *Psychological Review,* 104(4), 749-791.

Mitchell, C. M. 1987, GT-MSOCC: a research domain for modeling human-computer interaction and aiding decision making in supervisory control systems, *IEEE Transactions on Systems, Man, and Cybernetics*, 17(4), 553-572.

Mitchell, C. M., Rubin, K. S., and Govindaraj, T. 1986, OFMspert: An operator function model expert system, *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics,* 282-285.

Nairne, J. S. 1996, Short-term/Working memory, in E. L. Bjork and R. A. Bjork (eds), *Memory* (San Diego, CA: Academic Press), 237-313.

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

Newell, A., and Simon, H. A. 1972, *Human Problem Solving* (Englewood Cliffs, NJ: Prentice Hall).

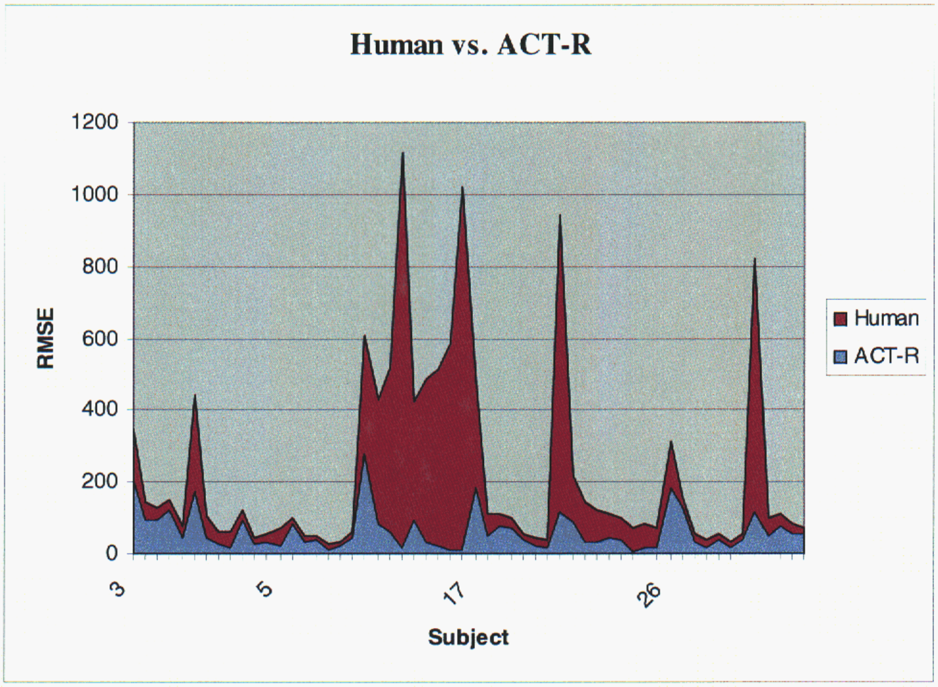
Pew, R. W., and Mavor, A. S. 1998, *Modeling Human and Organizational Behavior: Application to Military Simulations,* Panel on Modeling Human Behavior and Command Decision Making: Representation for Military Simulations (Washington, D.C.: National Academy Press).

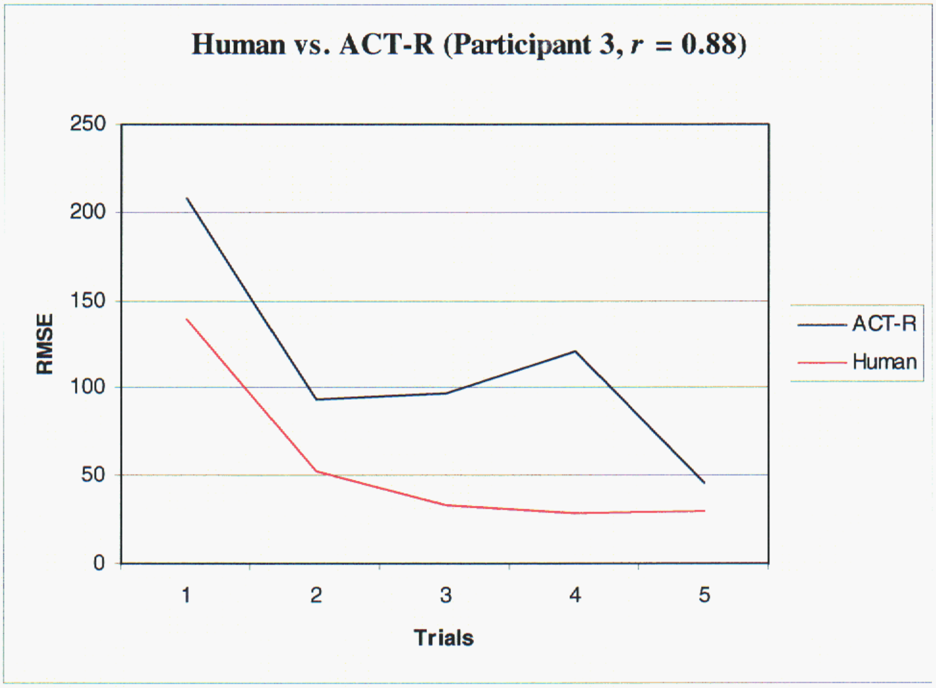
Ram, A., Narayanan, S., and Cox, M. T. 1995, Learning to troubleshoot: Multistrategy learning of diagnostic knowledge for a real-world problem-solving task, *Cognitive Science*, 19(3), 289-340.

Rosenbloom, P. S., Laird, J. E., and Newell, A. (Eds.), 1993, *The Soar Papers: Research on Integrated Intelligence* (vol. 1 – 2), (Cambridge, MA: MIT Press).

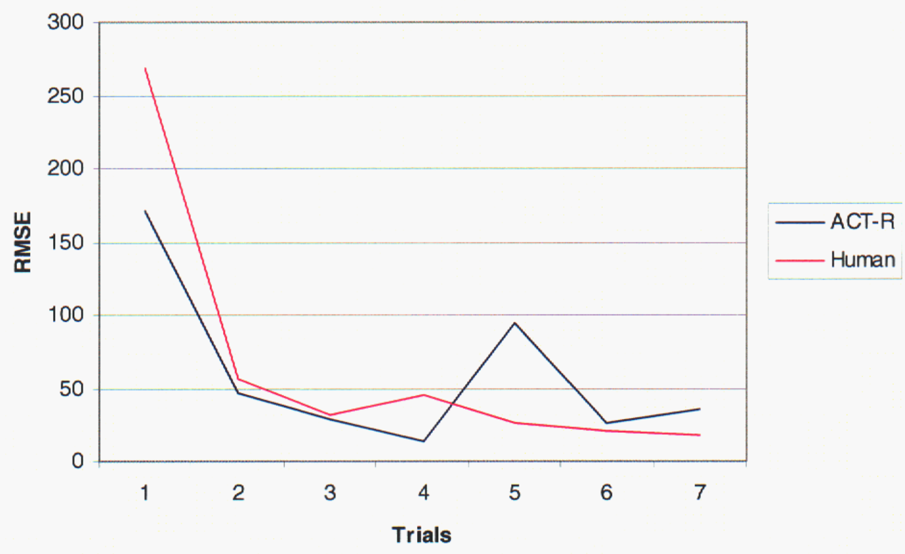
APPENDIX

Subject-to-Subject Comparisons of Human and ACT-R (Control Group) Performance

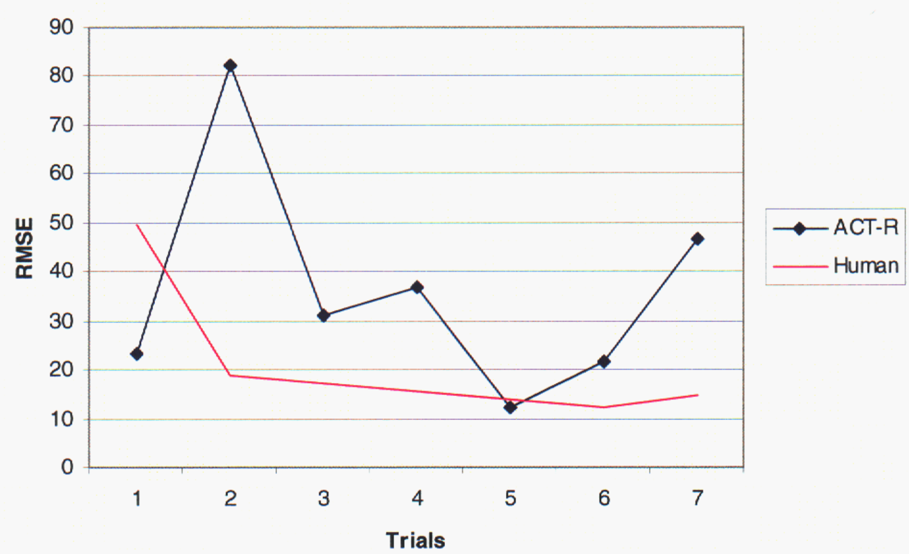


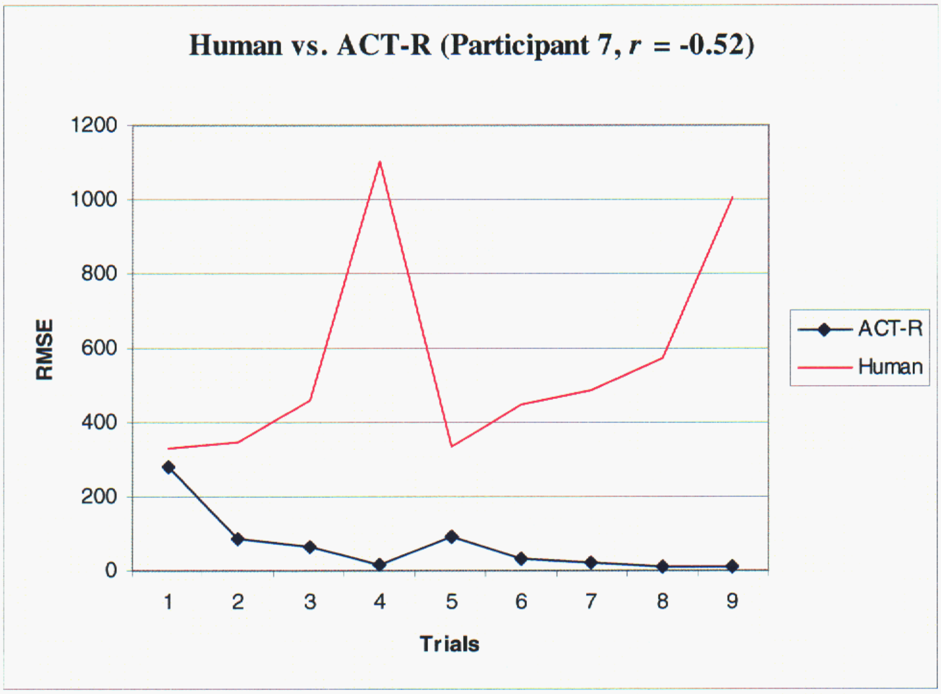


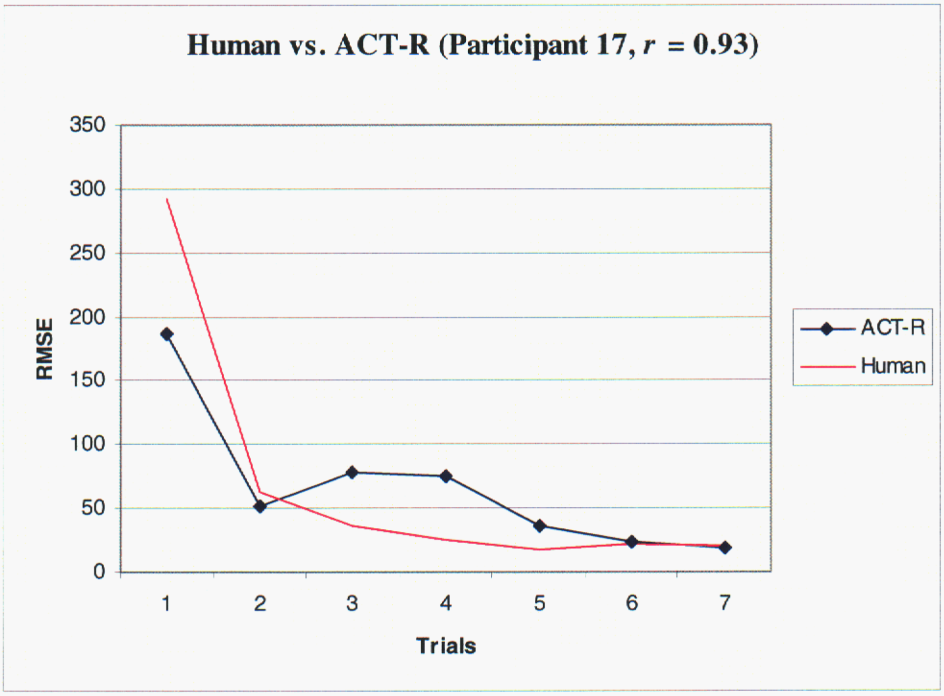
Human vs. ACT-R (Participant 4, r = 0.86)

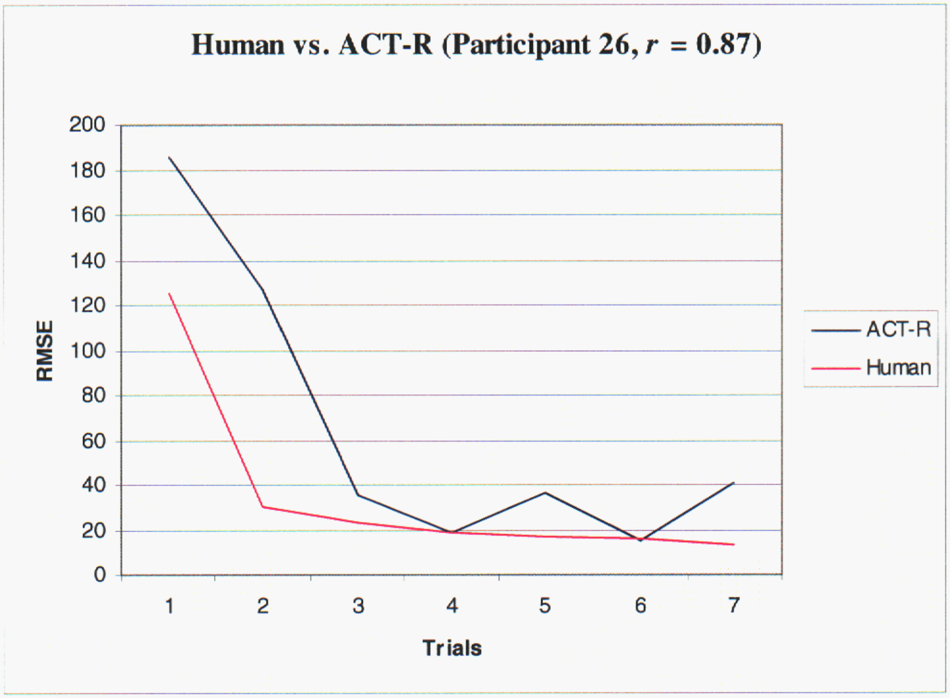


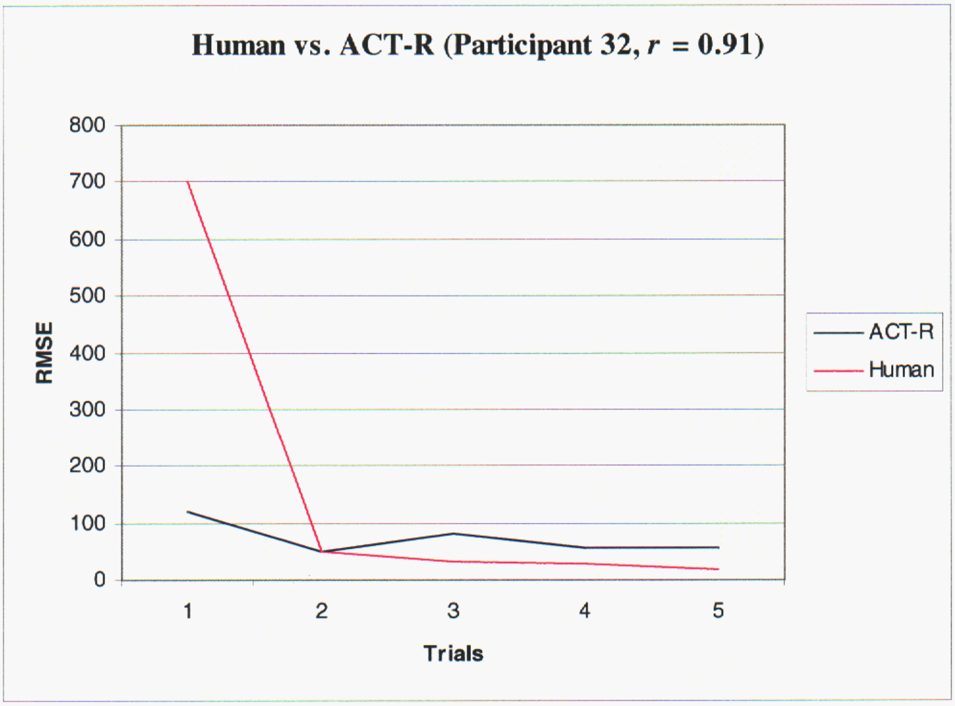
Human vs. ACT-R (Participant 5, r = -0.12)




Human vs. ACT-R (Participant 7, r = -0.52)
1200
1000
800
600
400
200
0
RMSE
1
2
3
4
5
6
7
8
9
Trials
ACT-R
Human


Human vs. ACT-R (Participant 17, r = 0.93)
350
300
250
200
150
100
50
0
RMSE
1
2
3
4
5
6
7
Trials
ACT-R
Human


Human vs. ACT-R (Participant 26, r = 0.87)
200
180
160
140
120
100
80
60
40
20
0
RMSE
1
2
3
4
5
6
7
Trials
ACT-R
Human


Human vs. ACT-R (Participant 32, r = 0.91)
800
700
600
500
400
300
200
100
0
RMSE
1
2
3
4
5
Trials
ACT-R
Human

Distribution:

1	MS0830	Nathan Brannon (12335)
1	MS0830	Kathleen Diegert (12335)
1	MS1188	J. Chris Forsythe (15311)
1	MS1004	Patrick Xavier (15221)
1	MS9018	Central Technical Files, 8945-1
2	MS0899	Technical Library, 9616
1	MS0612	Review & Approval Desk, 9612