Learning Under High Cognitive Workload

F. Javier Lerch (fl0c@andrew.cmu.edu)
Cleotilde Gonzalez (conzalez@andrew.cmu.edu)
Christian Lebiere (cl@andrew.cmu.edu)

Center for Interactive Simulations
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh PA 15213 USA

Abstract

This research investigates the impact of time pressure and individual differences on learning in a Real-Time Dynamic Decision Making (RTDDM) task. Our empirical results indicate that high time pressure generates high cognitive loads inhibiting learning. The results also show that high time pressure have a differential impact on the learning of individuals with high or low Working Memory (WM) capacity. We present a cognitive model based on ACT-R intended to explain learning in this task. Our cognitive model simulates learning by recognizing regularities in the decision task, and building "chunks" that guide decision making (instance-based learning). We describe how the model will be used to explain the impact of time pressure and WM capacity by varying the number of chunks acquired by the system given alternative time pressure conditions and individual differences.

Introduction

Real-Time Dynamic Decision Making (RTDDM) tasks have three main characteristics: a) the decision maker has to make a series of interdependent decisions; b) the environment changes because of exogenous events and because of prior decisions; and c) the pacing of decisions is dictated by the task rather than by the decision maker (Brehmer, 1990). This research investigates the impact of time pressure and individual differences on learning in a RTDDM task. It attempts to explain these phenomena by building a detailed cognitive model of the decision maker. The rationale for the cognitive model is to have a more in-depth understanding of why time pressure and individual differences foster or inhibit learning. We expect this detailed understanding would help us build better training and decision aids for RTDDM tasks.

Theory

In most RTDDM tasks the rules for making individual decisions are simple. For example, air traffic controllers need to identify if two airplanes are in a collision course. If this is the case, they need to ask one of the airplanes to change direction. But the tasks are rather complex because of the interdependency of decisions and the time pressure to make them. Under these conditions, we expect most learning will be instance-based learning. That is, decision makers will learn chunks expressing under which task conditions specific decisions have the desired effect on the system.

WM is the system for holding and manipulating information during the performance of cognitive tasks (Baddeley, 1990). Limitations in WM capacity have been recognized as a major bottleneck in human cognitive processing. We expect that differences in WM capacity will have a great impact on how individuals perform and learn in RTDDM tasks because these environments impose a high cognitive workload. More specifically, we expect that individuals with high WM resources will learn faster because they have the additional cognitive resources to reflect on the impact of their prior decisions, and to store more and better chunks. Also, since WM capacity is used for both performance and learning, we expect that decision makers will learn faster if they are first trained in a low time pressure environment, and then they are asked to make decisions in the higher time pressure environment. Conversely, individuals that are trained from the beginning in the high time pressure environment should find it harder to learn because all their cognitive resources are devoted to executing the task, and they have less spare resources devoted to learning. This prediction should be mediated by individual differences in WM.

WM is divided into two subsystem: 1) a linguistic sub-system, and 2) a spatial sub-system. In the linguistic sub-system, information is kept in linguistic code, and the processing can be characterized as sequential and propositional. In the spatial sub-system, information is kept in visual code, and the processing can be characterized as more parallel and analogical. There is strong evidence that language processing and spatial thinking are supported by separate pools of WM capacity (Shah and Miyake, 1996). Prior studies have shown that individuals with high linguistic WM capacity perform better than individuals with low linguistic WM capacity in a variety of real-time tasks such as reading comprehension (Just and Carpenter, 1992) and phone-based interaction (Huguenard, Lerch, Junker, Patz and Kass, 1997). Our RTDDM task is highly spatial so we expect that individuals with high spatial WM capacity will perform and learn better than individuals with low spatial WM capacity.

In our research we use traditional measurements of linguistic and spatial WM capacity. We also use the Raven Progressive Matrices Test (Raven, 1962) as an additional measurement of spatial WM. Prior research using detailed eye-tracking analysis has shown that differences in Raven tests can be explained by the ability to induce abstract spatial relations and the ability to dynamically manage a large set of problem-solving goals in WM (Carpenter, Just and Shell, 1990).

Laboratory Study

We used a simulation tool called Pipes. Pipes is an abstraction of a resource management task that can be performed by a single individual or a group. The task is an isomorph of a real-world task in an organization with large-scale logistical operations (the United States Postal Service). We have built a realistic simulation of the task, but this simulation is too complex and takes too long to learn to be practical in laboratory studies (See Lerch, Ballou and Harter, 1997 for a detailed description of the realistic simulation). On the other hand, Pipes can be learned in approximately one hour, and a complete trial can be run in few minutes. Pipes simulates a water distribution system (isomorph to mail sorting) with multiple deadlines for alternative tanks in the system. The whole simulation is spatial. Decision makers have to decide when to activate or de-activate pumps given that the number of pumps working at any given time is restricted (this is isomorphic to having a limited number of sorting machines in the USPS). The task is highly dynamic because water may arrive into a tank at any time, and the level of water in each tank depends on prior decisions (i.e., the pumps that were activated or de-activated by the decision maker in the past). The task is also real-time because pumps are activated or de-activated as the simulation clock is running (See Figure 1 for the main layout of the simulation).

We ran 33 participants using this simulation. Each participant was run in five consecutive days, and paid $50 at the end of the 5 days. In the first two days, each participant completed three psychological tests: the Reading Span Test (Daneman and Carpenter, 1980) that measures WM capacity for language processing; the Spatial Span Test (Shah and Miyake, 1996) that measures WM capacity for spatial thinking; and the Raven Progressive Matrices Test (Raven, 1962).

We manipulated time pressure by changing the speed of events. In the last three days, each participant was randomly assigned to one of two groups: the Fast-Fast (FF) condition and the Very Slow-Fast (VSF) condition. The exogenous events in all trials were identical. The simulation was run either in a Fast mode (8 minutes trials), or in a Very Slow mode (24 minutes trials). In the FF condition, participants ran the simulation 6 times for three days in the Fast mode (18 trials over three days). In the VSF condition, participants ran the simulation in the Very Slow mode for the first *two* days. In these *two* days, they only ran 2 trials per day, so their total time on task was the same as the time on task for Fast-Fast participants (Very Slow-Fast: 2 trials x 24 minutes = 48 minutes per day; Fast-Fast: 6 trials x 8 minutes = 48 minutes per day). In the third (last) day, the Very Slow-Fast participants ran the simulation 6 times in the Fast mode (8 minutes trials), the same as the Fast-Fast participants. We expect Very Slow-Fast (VSF) participants will exhibit more learning than Fast-Fast (FF) participants.

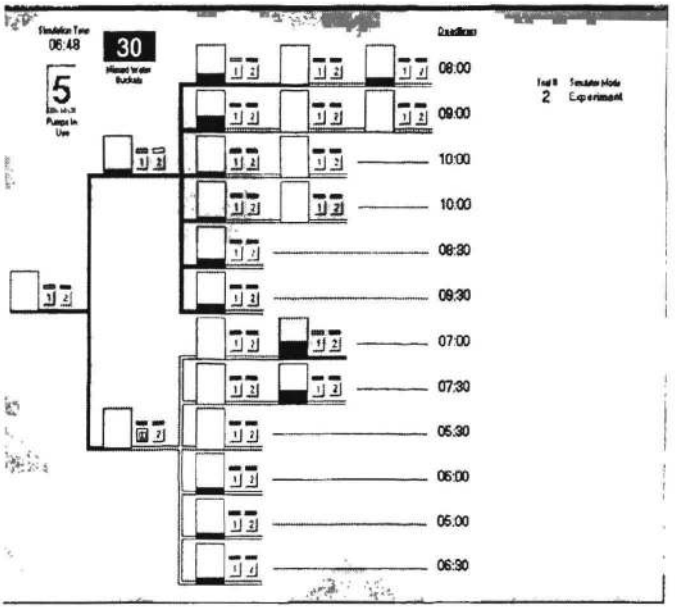

Figure 1. The Pipes simulation

Experimental Results

Figure 2 shows that all three measures of individual differences are correlated. Also, as expected, they show that Raven is more highly correlated to spatial WM capacity than to linguistic WM capacity.

We averaged the results of each participant across trials for each day so each participant had only three repeated performance measures (one for each day). Our performance measure is the number of water buckets that were not pumped in time, therefore the higher the number of water buckets missed, the worse the performance of the decision maker.

We first ran an analysis of variance with three repeated measures with only spatial WM as a covariate. The results show that individuals with high spatial WM capacity performed better than those with low spatial WM capacity [$F(1,29)=4.813$, $p=.036$)]. Second, we ran the same analysis with only linguistic WM capacity as a covariate. Linguistic WM capacity was not significant [$F(1,29)=.341$,ns]. We then ran the same analysis with spatial WM capacity and Raven as covariates. The results are shown in Figure 3.

	Raven	Linguistic WM	Spatial WM
Raven	1		
Linguistic WM	0.391	1	
Spatial WM	0.504	0.545	1

Figure 2. Correlations Among the Three Tests.

	Between-Subjects Effects				
	df	Mean Square	F	Sig	Observed Power
Condition	1	3638.892	0.879	0.357	0.148
Spatial WM	1	2140.571	0.517	0.478	0.107
Condition*Spatial WM	1	7816.475	1.888	0.181	0.263
RAVEN	1	29031.594	7.013	**0.013**	0.723
Condition*Raven	1	6710.012	1.621	0.214	0.233
Error	27	4139.791			
	Within-Subjects Effects				
	df	Mean Square	F	Sig	Observed Power
Trial	2	683.558	1.065	0.352	0.227
Trial*Condition	2	4587.988	7.149	**0.002**	0.919
Trial*Spatial WM	2	312.358	0.487	0.617	0.126
Trial*Condition*Spatial WM	2	515.578	0.803	0.453	0.180
Trial*RAVEN	2	186.382	0.290	0.749	0.094
Trial*Condition*RAVEN	2	4158.304	6.480	**0.003**	0.890
Error (Trial)	54	641.757			

Figure 3. Between and Within Subjects Effects

These results indicate that when both covariates are used only Raven is significant (p=.013). The analysis of variance also shows two significant interaction effects: Trial X Condition interaction (p=.002) and Trial X Condition X Raven interaction (p=.003).

Figure 4 shows the interaction between trial and time pressure. The graph shows that performance was very similar the first day between the FF and VSF participants. But VSF participants improved their performance faster than FF participants. It is important to remember that VSF participants only had 4 trials in the first two days (2 trials per day) while FF participants ran the simulation 12 times (6 times per day). All participants ran the simulation 6 times the last day under high time pressure.

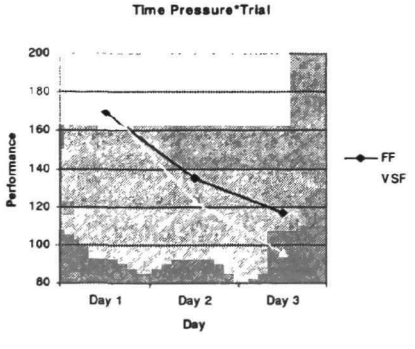

Figure 4. Trial - Time pressure interaction.

Finally, Figure 5 shows the triple interaction. The left panel shows the results of the Low Raven subjects (we divided subjects by using the mean of our sample). The graph shows that Low Raven subjects greatly benefited by first being trained in the low time pressure condition before being exposed to the high time pressure version of the simulation (VSF condition). Although Low Raven subjects in the FF condition performed better the first days than subjects in the VSF condition, they exhibited little learning throughout the three days.

The right panel graph shows the results for the High Raven subjects. In this case, the benefits of the VSF condition on learning and performance are very small throughout the 3 days. It also shows that High Raven subjects had a better performance than Low Raven subjects consistently (compare left and right panels).

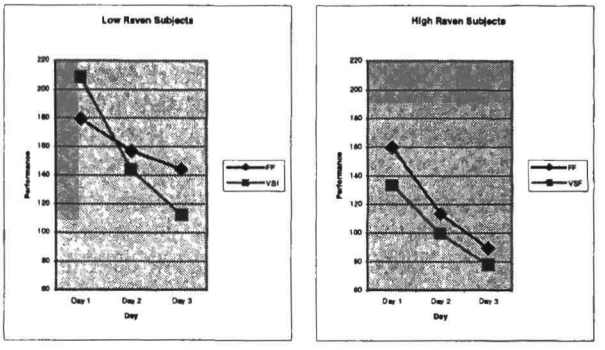

Figure 5. Triple interaction

What Was Learned?

Our next step was to analyze each decision made by each subject in order to figure out what subjects were learning. We hypothesized that subjects would learn chunks representing under which task conditions specific decisions improve performance rather than learning decision rules (or improving their implementation of these rules). In our analysis, we compared each decision in each trial (between 30 and 60 decision per trial) for each day (6 trials in the Fast condition and 2 trials in the Very-Slow condition) for each subject (33 participants) against standard decision rules. These rules were derived from the scheduling literature and the verbal protocols of pilot subjects. For example, a standard rule is the slack rule. In the slack rule you take into account the time left before the deadline and the volume of water in each tank, and select to activate the pump(s) of the tank with the lowest slack (we call this strategy the Time-Volume rule). We did this analysis using several decision rules. For each decision and for each decision rule we calculated a goodness of fit coefficient using the following formula:

Goodness of fit = 1 - ((slack - minimum) / (maximum - minimum))

This coefficient has values between 0 and 1 and represents the similarity between a decision rule and each decision made by the subject. A coefficient fit of 1 means perfect

agreement (i.e., the slack of the subject's decision is the same as the minimum slack in the environment). In such a case the subject has chosen the best decision according to the Time-Volume rule. On the other hand, a coefficient fit of 0 is equivalent to the subject selecting the maximum slack in the environment. In this case, the subject has chosen the worst decision according to this rule. In this paper we only report the results of the fit for the Time-Volume rule since the results are similar for the other rules (and because of space constraints).

Figure 6 shows the results of the average fit of all decisions for each day (several trials per day) across all subjects in the FF and VSF conditions. The graph shows that the rule fit *declines* through time, that is, subjects follow the rule less as they are learning. It also shows that VSF subjects (the best learners) had a more pronounced decline in their rule-following fit. Similar declines were found for simpler and for more complex rules. Those subjects that learn the most are those that learn to follow the standard rules less often. These subjects seem to make decisions by being more data driven, that is, by exploiting specific task conditions and making decisions that may have worked in the past.

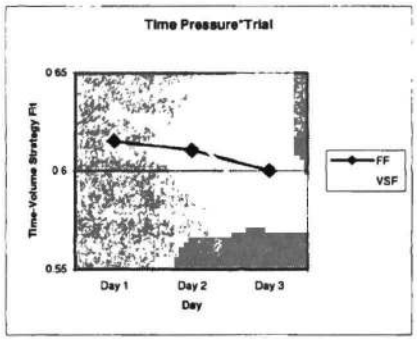

Figure 6. Average fit of all decisions per day

Figure 7 shows the results of the average fit for the Time-Volume decision rule for Low and High Raven subjects. We would expect that Low Raven subjects may be less able to exploit the specific conditions of the task environment because they have less cognitive resources to analyze and store chunks on what decisions worked under which task conditions, especially if they are trained in the high time pressure simulation (FF condition). The left panel shows that Low Raven subjects in the FF condition in fact *increase* their fit to the Time-Volume rule from day 1 to day 3. These are the subjects who exhibited the worst learning. On the other hand, Low Raven subjects in the VSF condition started with a high fit coefficient but lower this coefficient through time. The right panel shows the results for the High Raven subjects. Their rule fit coefficients were lower than for the Low Raven subjects, and decline through time for both experimental conditions.

Our hypothesis here is that subjects that are more data driven should perform better and learn more. If this is true, then we should expect that the best performers were not only those that followed the rule the least, but are also more adaptive. One measure of adaptation is the standard deviation of the goodness of fit coefficients *within* a trial. Two subjects may have the same average goodness of fit coefficient in a trial, but very different standard deviations. The subject with the high standard deviation follows the rule very closely for some decisions, and not all for others, while the subject with the low standard deviation follows the rule at the same level for all decisions. To test this hypothesis we ran a regression of performance for each trial (450 trials for all subjects)against the following variables:

a) Raven score
b) Average rule fit per trial
c) Standard deviation of rule fit per trial
d) Two other measurements of how well subjects used the task environments resources (i.e., pump time)

The results were highly significant (Adjusted R^2 = .785). There are no co-linearity problems among the explanatory variables. The highest standardized coefficients were for the two measurements of resource utilization (-.647 and -.320; negative coefficients mean performance improvements). The standardized coefficients for the other three variables are: Raven = -.161 ($p<.001$), Average fit = .114 ($p=.015$), and Standard Deviation of fit = -.208 ($p<.001$). These coefficients indicate that subjects with higher Raven scores have better performance, trials with a higher average rule fit have worse performance (after controlling for Raven score and resource utilization), and finally, trials with higher standard deviation have higher performance. The last coefficient suggests that the less consistent subjects are following the Time-Volume rule (after controlling for average fit), the better their performance. Similar results apply to other decision rules. These results indicate that data driven decision making is beneficial in this task environment.

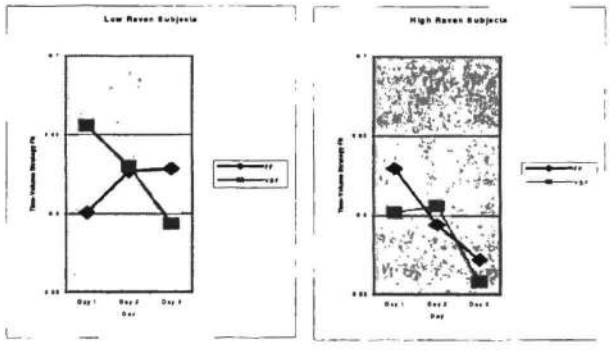

Figure 7. Average fit for Time-Volume Rule for Low and High Raven.

The Act-R Theory

ACT-R is a theory of cognition that has been applied to a wide range of cognitive tasks since its introduction in 1993 (Anderson and Lebiere, 1998). ACT-R assumes two types of memory: procedural and declarative. Procedural memory contains skills in the form of productions or rules of action.

Declarative memory holds explicit knowledge represented as chunks. Production rules specify how the chunks are used to solve problems. ACT-R is a goal-directed architecture. At each cycle, one goal is designated as the top goal or focus of attention. A production is then selected that matches that goal, retrieves a chunk from memory (if necessary), then transforms the goal. This is a symbolic description of ACT-R in terms of how productions and chunks interact.

ACT-R has also a sub-symbolic level. At this level, ACT-R provides real-valued quantities associated with declarative and procedural knowledge to produce a more accurate picture of the graduated nature of human cognition. Their purpose is to resolve conflicts: when several productions match the current goal or several chunks match a production condition, the sub-symbolic parameters associated with those symbolic structures will determine which is selected, and how quickly. Those parameters are learned to optimize the model to the structure of the environment.

In this model, we will concentrate on the acquisition and use of declarative knowledge. We will assume that the productions used to manipulate those chunks, and their parameters, reflect some general, well-established knowledge on how to solve problems of this type.

For declarative knowledge, a chunk is defined as a collection of slots, each of which can hold another chunk as value, and is associated to a quantity called activation which represents the chunk's history of use and determines its availability. Specifically, the activation A_i of chunk i is defined by the formula:

$$A_i = B_i + \sum_j W_j \cdot S_{ji} + N(0,\sigma)$$

B_i is called the base-level activation. It increases with the number of references to the chunk (practice) and decays with time (forgetting). The second term represents the activation spread from each source j according to its source level W_j and its strength of association, S_{ji}, to the chunk. The values of the current goal are the sources of activation, which evenly share a total source amount W. Finally, zero-mean Gaussian noise of standard deviation σ is added to the activation.

In a task such as this featuring continuously evolving quantities such as time and amount of water, no match to declarative memory is ever likely to be perfect because no situation is ever encountered precisely the same way twice. A mechanism called partial matching allows a chunk to be retrieved even if it only partially matches a production condition. A quantity called the match score of chunk i to production p is defined by subtracting from the chunk activation an amount proportional to the degree of mismatch:

$$M_{ip} = A_i - MP \cdot \sum_{v,d} (1 - Sim(v,d))$$

MP is a scaling parameter called the mismatch penalty and Sim(v,d) is the similarity between each production condition d and corresponding chunk value v. The chunk with the highest match score will be retrieved from memory if its score is above the activation threshold τ. Otherwise, the production fails and another is selected. Finally, the latency to retrieve a chunk is inversely proportional to its match score, making more active, better-matching chunks faster to access. The addition of noise to the activation makes declarative retrieval a probabilistic process, and the mechanism of partial matching makes it an approximate process. Stochasticity and generalization are two human qualities in constant display in this task.

These mechanisms of ACT-R can be used to implement a "user model" of the task that will generate detailed predictions for each action and its latency at every step of the problem-solving process.

An ACT-R Model of the Pipes RTDDM Task.

ACT-R has successfully modeled phenomena of memory, problem solving and skill acquisition (Anderson and Lebiere, 1998). However, most of the tasks modeled up to now are static, relatively simple tasks. Recently, there has been more interest in modeling more dynamic and complex tasks in ACT-R. Figure 8 represents our proposed ACT-R model for the Pipes task. The overall goal is represented as a set of deadline chunks. The focus of attention is the deadline closest to the current simulation time.

Declarative memory has two main chunk structures: "the tank" and "the decision." The information provided in the tank chunk corresponds to the physical representation of a tank in the system, and what the user is aware of: the water amount, the deadline, the connections with other tanks, and the status of the pumps in that tank. The chunk called "decision" stores the information on the evaluations performed on the tanks during the course of the learning process, namely water amount, time until the deadline and evaluation. The first time the model is run, no decision chunks exist. Decision chunks are created in the course of solving the current problem: when a goal that was set to evaluate a tank is completed, it becomes a declarative memory chunk holding the information relevant to the evaluation. If the same evaluations are made or retrieved in future trials, the decision chunk increases its activation value, increasing the probability of being retrieved in the future. Decisions are updated according to the feedback provided by the system (i.e., no missed buckets decreases the evaluation because more slack was available whereas a high number of missed buckets increases the evaluation that generated this situation because the tank should have been given a higher priority). Since an identical situation is unlikely to occur, the partial matching mechanism provides a certain amount of generalization in finding the "correct" decision, i.e. a particular decision chunk may be retrieved if the time until deadline and amount of water in the tank is close enough to the current situation.

Procedural memory consists of 5 basic activities: evaluate tanks for which pumps may be turned on or off, turn specific pumps on or off, and review the environment (e.g., re-start the evaluation cycle). When the user evaluates the tanks, the

model assumes that the user will keep a value of "urgency" for each tank and type of action (urgency to turn-pumps associated with each tank on or off) in the chunks. Two productions are available to evaluate a tank. The first one will try to retrieve a prior decision closely matching the characteristics of this tank (water, deadline). If no prior decision is sufficiently active and matches closely enough to reach the retrieval threshold, then that production will fail. The second production then will be selected that evaluates the chunk according to some general heuristic function. After all the tanks have been evaluated, the user then decides to turn on the pump associated to the most urgent tank if a pump is available, or to turn off the pump associated to the least urgent tank, thereby freeing a pump, assuming that the urgency of that tank is significantly less than that of the tank that needs to be turned on. Actions in the user model modify the status of the environment, which is updated by the simulation. According to the definition of RTDDM, the environment also changes independently from user's actions. The system adds water to the tanks, pumps water from previous tanks, automatically turns off pumps that correspond to tanks with no more water, and turns on pumps that have been queued by the user. The simulation also verifies the deadlines to provide feedback to the user.

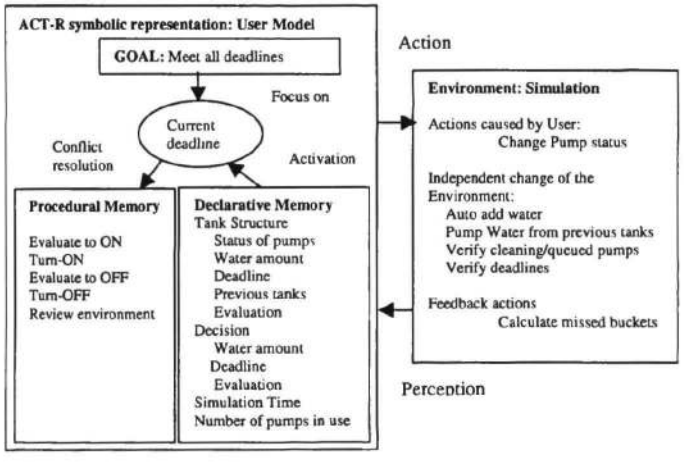

Figure 8. ACT-R Model for Pipes

Although very little work has been done in ACT-R to model individual differences, work by Lovett, Reder, and Lebiere (1999) indicate that the *W* parameter may be manipulated to capture individual differences in WM. However, that parameter controls the spreading activation component, which is essential to accounts of memory phenomena such as the fan effect, but not particularly relevant in this model. Therefore, we will also investigate if other parameters can account for individual differences, including the decay rate of base-level activation *d*, the mismatch penalty *MP*, the retrieval threshold τ and the activation noise magnitude σ. All of these parameters affect the activation that controls the availability of decisions chunks, but they act on separate components of the activation and thus are expected to exhibit different effects.

Time pressure in the model is implemented by modifying the rate at which the environment changes, and comparing the ACT-R's time to that rate. If the rate of change is very low (no time pressure), the user model will have time to complete more evaluations before the environment changes, and to better reflect and update its evaluations following system feedback. If the rate of change is very high, the user may not have time to evaluate the environment completely before it changes again, or to update its evaluations.

Conclusions

This research suggests that learning in real-time dynamic decision tasks depends on the spare WM resources available during task execution. It also suggests that most learning is based on acquiring relevant decision instances that exploit the task environment.

Acknowledgments

The research reported here was partially supported by a grant from the Air Force Office of Scientific Research (F49620-97-1-0368).

References

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

Baddeley A. D. (1990). *Working Memory: Theory and Practice.* Boston: Allyn and Bacon.

Brehmer, B. (1990). Strategies in Real-Time, Dynamic Decision Making. In R.M. Hogarth (Ed.), *Insights in Decision Making,* University of Chicago Press, 262-279.

Carpenter P.A., Just M.A., Shell P. (1990). What One Intelligence Test Measures: A Theoretical Account of the Processing in the Raven Progressive Matrices Test. *Psychological Review*, 97,3,404-431.

Daneman, M. and Carpenter, P.A. (1980) Individual Differences in Working Memory and Reading. *Journal of Verbal learning and Verbal Behavior.* 19, 450-466.

Huguenard B. R., Lerch F.J., Junker B.W., Patz R.J. and Kass R.E. Working-Memory Failure in Phone-Based Interaction (1997). *ACM Transactions on Computer-Human Interaction*, 4,2, 67-102.

Just, M.A. and Carpenter, P.A. (1992) A Capacity Theory of Comprehension: Individual differences in Working Memory. *Psychology Review.* 99, 1, 122-149.

Lerch, F.J., Ballou, D.J. and Harter, D.E. (1997). Using Simulation Based Experiments for Software Requirements Engineering. *Annals of Software Engineering,* 3, 345-366.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling Working Memory in a Unified Architecture: An ACT-R Perspective. In Miyake, A. and Shah, P. (Eds.) Models of Working Memory: Mechanisms of Active Maintenance and Executive Control. New York: Cambridge University Press, 1999.

Raven J.C. Advanced Progressive Matrices test. (1962) (distributed in the US by the Psychological Corporation).

Shah P. and Miyake A. (1996). The Separability of Working Memory Resources for Spatial Thinking and Language Processing: An Individual Differences Approach. *Journal of Experimental Psychology,* 125, 4-27.