Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation, Sundance, UT, 31 March - 2 April 2009
[paper: 09-BRIMS-010, pp. 39-46]

A High-Performance Approach to Model Calibration and Validation

Sue E. Kase
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16801-3857
skase@ist.psu.edu

Frank E. Ritter
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16801-3857
frank.ritter@psu.edu

Keywords: model validation, cognitive models, behavior moderators, genetic algorithms

ABSTRACT: *A new model validation approach is presented that integrates parallel processing on high-performance computing clusters with random search algorithms to fit cognitive models to human performance data. The efficiency, accuracy, and non-biasness of this approach surpasses the prevalent manual optimization techniques; results in exceptional model to human data fits; and is available and extendable to other parameterized models, search algorithms, cognitive architectures, and cluster computing resources. Results from testing the validation approach using a prototype cognitive model of a serial subtraction task, the ACT-R cognitive architecture, and 15 individual fits are described.*

1. Introduction

Model validation is an essential aspect of model creation and use. It is much more than the simple comparison of model predictions with empirical data and the binary determination that the model is or is not valid (Glenn, Neville, Stokes, & Ryder, 2004). Glenn et al. describe a continuum of validation processes for human performance models. At one end of the continuum is model *calibration* using the discrepancies between actual model predictions and empirical data to adjust parametric aspects of the model to improve correspondence for subsequent execution of the same model. For example, a parameter in the ACT-R cognitive architecture called *W* representing the sum of activations of all pieces of information in declarative memory was varied to model individual differences in working memory capacity in a digit memory task (Lovett, Reder, & Lebiere, 1997). At the other end of the validation continuum is fundamental inquiry regarding the inherent value of different modeling frameworks, paradigms, and philosophies. For example, in the Agent-based Modeling and Behavior Representation (AMBER) program (Gluck & Pew, 2001), a set of alternative knowledge-based cognitive architectures (ACT-R, Soar/Epic, DCOG, and iGEN) were compared for relative effectiveness in the simulation of human performance in the context of a simplified air traffic control task.

In Sargent's (2005) examination of various validation techniques, the terminology *parameter variability* and *sensitivity analysis* is used instead of calibration. Sargent describes this technique as changing the values of input and internal parameters of a model to determine the effect upon the model's behavior or output. Parameters identified as sensitive cause significant changes in the model's behavior or output. The same relationship should occur in both the model and what Sargent calls the real system.

Roberts and Pashler (2000) noted that the psychological research literature probably contained thousands of examples beginning in the early 1900s of quantitative psychological theories with free parameters supported by demonstrations that they can *fit* data—that the parameters can be adjusted so that the output of the theory resembles empirical results. This similarity is often shown by a graph with two functions: one labeled observed (or data), and the other labeled predicted (or theory). Roberts and Pashler argued that when the theory fits the data then the theory should be taken 'seriously'.

Similar terminology can be found in the portion of the human performance modeling community that focuses on knowledge-based models of cognitive performance. The terminology *fitting the model* is generally used when referring to the process of validating specific model configurations in detailed contexts by adjusting model parameters to achieve a valid model in a given instance of use.

Originating from components of previous studies spanning more than a decade (Ritter, 1991; Tor & Ritter, 2004), the high performance computing (HPC) and parallel genetic algorithm (PGA) validation

approach presented in this paper lies at the calibration end of Glenn's (2004) validation continuum. This validation approach is a precursor to more accurate, time and resource efficient methods and tools for the validation of human performance models developed within complex cognitive architecture environments.

Recent related research by Gluck, Scheutz, Gunzelmann, Harris, and Kershner (2007) examined large-scale computing resources for execution of cognitive model parameter sweeps, as well as possible architectures enabling volunteer computing environments. Raymond, Fornberg, Buck-Gengler, Healy, and Bourne (2008) employed genetic algorithms and simulated annealing written in Matlab to search the parameter space of an IMPRINT model. The research discussed in this paper integrates these two perspectives by offering a parametric search algorithm validation approach on a HPC platform.

The remainder of the paper is organized as follows: Section 2 is a brief description of the ACT-R cognitive architecture; Section 3 describes the most prevalent method of validation used by the cognitive modeling community; Section 4 defines the components of the HPC and PGA validation approach; Sections 5 and 6 discuss testing this validation approach by fitting a prototype cognitive model of a serial subtraction task and analyzing the results; and Section 7 presents concluding comments.

2. Cognitive Architectures

Many instances of cognitive architectures exist, for example: ACT-R (Anderson, 1993), Soar (Newell, 1990), and Epic (Meyer & Kieras, 1997). This research utilizes the ACT-R version 6.0 architecture. ACT-R is a two-layer modular cognitive architecture on a production system framework. One layer contains symbolic representations and has a serial flow in that only one production can fire at a time. The second layer is a sub-symbolic layer whose representations are numeric quantities that are the result of computations performed as if they were executed in parallel. In ACT-R cognition emerges through the interaction of a number of independent modules through buffers that can hold a declarative memory fact. Each of these modules is associated with specific brain regions and theories about the internal processes of these modules (Anderson, 2007). Figure 1 is a diagram of the modules and buffers making up the ACT-R cognitive architecture.

Declarative and procedural knowledge are symbolic level structures in ACT-R. Declarative memory contains chunks that are typed slot-value objects representing facts. Procedural memory consists of condition-action rules called productions. In most ACT-R models much of the quantitative structure of cognition is at the sub-symbolic level. Declarative memory chunks have a numeric activation value that is determined by the recency and frequency of use of the chunk and a component that reflects retrieval noise. Productions request the retrieval of the chunk from declarative memory that has the highest activation among all chunks that match a specified retrieval pattern. Productions have conditional constraints that are matched against the contents of the buffers. The production to execute is determined at the sub-symbolic level by calculating a utility value for each matched production. The production with the highest utility is executed which consists of performing the operations specified in its actions.

ACT-R offers many parameters to manipulate and adjust the sub-symbolic processes with each parameter having a meaning associated with a specific module and sub-symbolic process. Adjusting the values of the parameters adjusts the architecture's theory of cognition for the purpose of modeling cognitive aspects of a task.

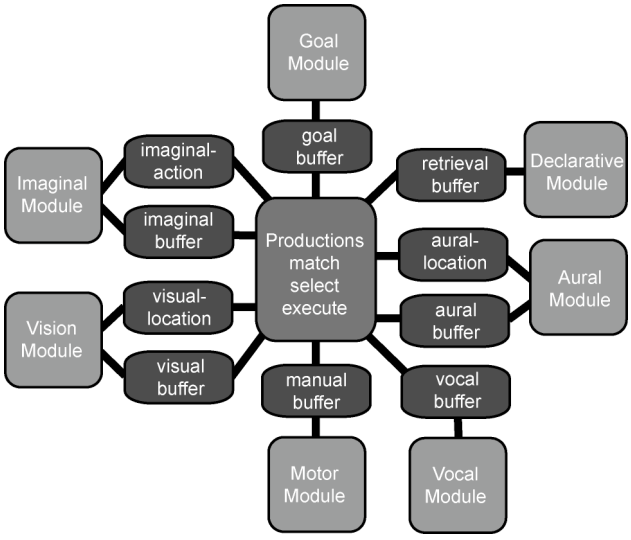

Figure 1: The ACT-R 6.0 cognitive architecture.

3. Traditional Validation Approach

If a cognitive model is structurally correct in simulating how a task is procedurally performed by humans, then the manipulation of architectural parameters can be used to simulate realistic effects on the cognitive performance of the task. For example, different sets of parameters and values can represent external conditions (e.g., conditions of the task environment, loud noise, human interruptions, or visual

distractions), or internal conditions pertaining to psychological or cognitive states of the humans performing the task (e.g., stressed, fatigued, or frightened). There has been a long tradition in the development of models for factors such as those listed above that appear to have moderating effects on virtually all human performance (Neville, Takamoto, French, Hursh, & Schiflett, 2000; Ritter, Reifers, Schoelles, & Klein, 2007). These types of moderating factors are sometimes called behavior moderators.

Traditionally, cognitive modeling researchers employ a manual optimization process to fit the model to human data. This is often a time consuming, iterative process involving the selection of architectural parameters supporting theory or a hypothesis, and then the assignment of a numeric value to each parameter. After the identification of parameters and value assignment, the model is run in the cognitive architecture and the resulting model predictions are compared to the human data. If the model fit is unsatisfactory, this fitting process is repeated and continues in a trial and error like manner building up a history of results that guide the next fitting attempt.

Manual optimization can be a reasonably effective process when using one parameter and fitting to one set of performance data, such as an average performance across subjects. However, when the goal is to fit a model to many individual subject's performance data, for example in the case of moderator factors or individual differences modeling, the validation effort in terms of time and computational resources can become prohibitively expensive on a single processor computer. Furthermore, systematic techniques for the estimation of multiple parameter value combinations out of the many free parameters available in architectures are notably absent in the modeling community.

When validation efforts associated with the manual optimization process impeded an investigation of how stress moderates cognitive performance (Ritter, Schoelles, Klein, & Kase, 2007), a new validation approach was developed for fitting a cognitive model of a serial subtraction task to human performance data from individual subjects.

4. New Validation Approach

The new validation approach is based on a genetic algorithm (GA) executed on a high performance cluster platform. This approach conducts an automated search of the cognitive model's parameter space for the best fitting combinations of parameter values that produce predictions that match human performance data.

GAs are based on the principles of natural selection and genetics, and have been applied successfully to numerous problems in business, engineering, and science (Goldberg, 1994). GAs are randomized, parallel search algorithms that search from a population of points. The points (often referred to as genotypes) represent individuals in a population. The genotypes are evaluated for fitness, then propagated to later generations by means of probabilistic selection, crossover, and mutation operations.

In a cognitive modeling context, the GA's genotypes are sets of cognitive architecture parameters applied to the cognitive model. The population evolves to find better 'solutions' by selecting the most fit parameter sets (those that give the best match to the human data), and propagating these solutions to the next generation.

In this validation approach the fitness evaluation consists of running the model in the cognitive architecture, analyzing the model's performance output, and calculating a fitness value for the model's predictions. This is done by running copies of the model—one per processor. A parallel version of the GA (PGA) distributes the computational load of the fitness evaluation among multiple processors reducing the time required to reach acceptable solutions.

There are several classes of PGAs distinguished by their level of parallelization (Cantu-Puz, 2001). This validation approach utilizes a master-slave global parallelization PGA. In a master-slave PGA, one master-processing node executes the GA-related functions, while the fitness evaluation is distributed among numerous slave processors. The slave processors evaluate the fitness of the genotypes that they receive from the master process, and then return the fitness results back to the master node. Table 1 presents pseudo code for optimizing a cognitive model using a master-slave PGA with a message-passing interface (MPI).

In the PGA, the slave processors each receive a different set of ACT-R architecture parameters representing a genotype, run the cognitive model in the architecture, collect the model output, and calculate the associated statistics and fitness value based on the model's performance compared to the human data.

In this case, the ACT-R architecture and cognitive model are written in the Lisp programming language. For this project the CMUCL dialect was used because it runs on most Unix platforms. Generally, MPI is available on cluster computing resources in the form of C or Fortran libraries. To utilize parallel processing in the cognitive model fitting process, ACT-R and the cognitive model are packaged into an executable Lisp

image file. This image file can be run by a system call from a C program on each processor in parallel while utilizing MPI to communicate genotypes and fitness values among the processors. Figure 2 illustrates the components of the validation approach executed on a high-performance cluster located at the National Center for Supercomputing Applications (NCSA).

Table 1: Pseudo code for master-slave PGA using MPI.

```
MPI_Init . . .
if (rank is 0)  // master
    Initialize population
. . . . .
for (each generation)
{
    if (rank is 0)  // master
    {
            Selection
            Crossover
            Mutation
            // creates a new generation
    }

    // find fitness of genotypes in population
    // master and slaves
    MPI_Scatter individuals out to processors
    Run cognitive model
    Calculate fitness of model predictions
    MPI_Gather up resulting fitness values

    if (rank is 0)  // master
            Print out generational statistics
}
Test best solutions found // master and slaves
MPI_Finalize . . .
```

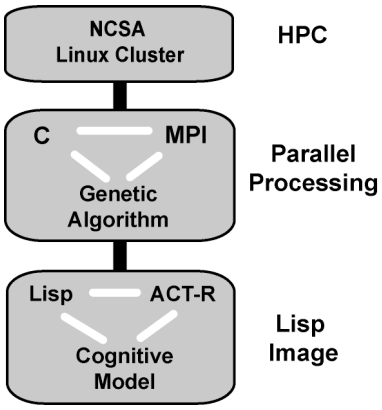

Figure 2: Components of HPC and PGA validation approach executed on a Linux cluster at NCSA.

The population of genotypes (ACT-R parameter sets), in the form of a matrix, are 'scattered' row-wise to the processors. Each processor executes the Lisp image file that runs the model within the ACT-R architecture. Each processor then calculates a fitness based on the model's performance predictions and the human data statistics. In this case, sum of the squares error is calculated on two performance statistics. The fitness values calculated by the processors are 'gathered' up by the master process, which then applies genetic functions to the population based on the fitness of the genotypes (refer to Table 1). This is repeated through a number of generations with the effect of evolving a set of candidate solutions.

5. Testing the Validation Approach

The HPC and PGA validation approach was tested by fitting a prototype cognitive model of the serial subtraction task written in the ACT-R cognitive architecture to individual-level performance data. The model was developed in collaboration with Dr. Michael Schoelles at the Cognitive Science Department at Rensselaer Polytechnic Institute (Ritter et al., 2007). The model simulates a human subject performing the serial subtraction task. Serial subtraction is the mental arithmetic stressor portion of the Trier Social Stressor Test (TSST, Kirschbaum, Pirke, & Hellhammer, 1993). The TSST has been used to provide an acute physiological stress response in human subjects in 100's of studies since the 1960's.

The serial subtraction task consisted of four 4-minute blocks of mentally subtracting by 7s and 13s from 4-digit starting numbers. Figure 3 illustrates the serial subtraction task with the four starting numbers for each subtraction block shaded in gray. The task is performed mentally with no visual or paper clues. An experimenter gives the subject the starting number; from then on, the subject speaks the answer to each subtraction problem.

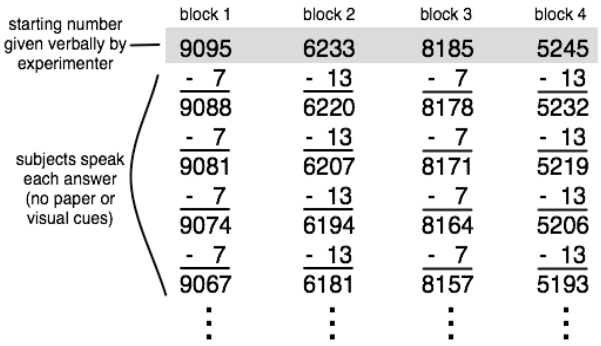

Figure 3: An illustration of the four blocks of the serial subtraction task as in the experiment; subjects perform the task mentally without paper or visual cues.

During the serial subtraction task, subjects' answers were scored against a list of correct answers from the starting number. For each subject the number of subtraction problem attempts was recorded and a percent correct score was calculated by dividing the

total number of correct attempts by the total number of attempts for each block of the subtractions.

Table 2 shows the subtraction rates for the subjects' performance on the two blocks of subtracting by 7s. The large standard deviations indicate a wide range of performance on this task suggesting a high degree of individual differences within the subject pool.

Table 2: Human subject (N=15) mean performance and standard deviation for serial subtraction on two 4-minute blocks of subtracting by 7s.

	7s – 1st block	7s – 2nd block
Number of Attempts	47.3 (15.2)	47.8 (19.2)
Percent Correct	82.0 (10.0)	88.8 (7.0)

To test the validation approach, 15 PGAs were setup to fit the serial subtraction model to each individual subject's performance data. Each PGA ran 100 generations of 200 binary-encoded genotypes. The genotypes consisting of three 12-bit substrings each representing the value of an ACT-R parameter. This means that one run of the PGA would sample 20,000 parameter combinations. Three ACT-R parameters appeared important for the model and were incorporated into the genotype's composition: seconds-per-syllable (SYL), base level constant (BLC), and activation noise (ANS).

The model's speaking rate is controlled by the SYL parameter. The ACT-R default timing for speech is 0.15 seconds per assumed syllable based on the length of the text string to speak. There is a default of three characters per syllable controlled by the characters-per-syllable parameter. The seconds-per-syllable and characters-per-syllable parameters control sub-symbolic processes in ACT-R's vocal module. The vocal module gives ACT-R a rudimentary ability to represent speech and the time and content of speaking.

The other two parameters making up the genotype (BLC and ANS) affect declarative knowledge access. The BLC parameter and a decay parameter affect declarative memory retrieval and retrieval time. The ANS value affects variance in retrieving declarative information and error rate for retrievals in the model. Other parameters, such as base level learning, decay, and the characters-per-syllable parameters were built into the model as modifiable but were left at their default values throughout the PGA runs. The search space for this optimization problem was defined by the following parameter value boundaries: both ANS and SYL 0.1 to 0.90 and BLC 0.1 to 3.0.

In the PGA code the selection probability (selection of the fittest) was set to 0.5, meaning half the genotype population is replaced each generation by offspring of the fittest genotypes. Random mutations alter a certain percentage of the bits in the list of genotypes. This operation introduces new traits in the original population and keeps the PGA from converging too quickly before sampling the entire search space. The mutation rate was set at 0.15. The terminating condition was a specified number of generations (100), instead of proximity to performance statistics.

The PGA's fitness function compared the sum of the squares error for the model's predicted number of attempted subtractions and percent correct to the corresponding human data. For these individual-level problems, the *fitness is in terms of error* (or cost) and is the discrepancy between the model's predictions and the actual human performance on the cognitive task (e.g., $(83\text{-}83.2)^2 + (94\text{-}94.2)^2 = 0.1463$). Therefore, a fitness value of zero means there is no discrepancy between model predictions and human data—the model predicted the human performance perfectly. The PGA produced fitness values are discussed next.

6. Validation Results

This section examines the results of fitting the serial subtraction model to performance data from the 15 subjects in the experiment using the HPC and PGA validation approach. Table 3 is a summary of the results for the subjects ordered by human performance on number of attempts from worst to best performance.

The fitness value column shows good fits. Genotypes resulting in fitness values close to zero were found for all subjects. The closer the fitness value is to zero the better the model-to-data fit. Fitness values ranged from 0.0006 to 0.7682. When comparing the human performance column to the model prediction column, both number of attempts and percent correct were fit to the human data to within a fractional part of a subtraction problem and a fractional part of a percentage point for all subjects. The best/lowest fitness value (0.0006) in Table 3 corresponds to the subject with the worst performance by number of attempts (subject 1). There is a cluster of excellent fits, the first three rows in Table 3, for the poor performers (0.0006, 0.0866, 0.0487). In the next three rows, there is a cluster of fits closer to 1.0 (0.6836, 0.5523, 0.7682), with row 6, subject 2's fit, producing the worst fitness value (0.7682) out of all the subjects.

The genotype column in Table 3 shows the ACT-R parameters values in the sequence of ANS, BLC, and SYL that produced the best fit for each subject. By

examining the contents of Table 3 trends or patterns in reference to parameter values and performance can be observed for good fits, and represent changes in the mechanisms of cognition between these subjects.

Table 3: HPC and PGA validation results for 15 subjects comparing human performance (number of attempts and percent correct), model predictions (number of attempts and percent correct), and fitness values with the corresponding genotypes (ACT-R parameters ANS, BLC, SYL).

Subject	Human Performance	Model Prediction	Fitness Value	Genotype ACT-R Parameters ANS, BLC, SYL
1	28, 67.9	28.0, 67.8	0.0006	0.83, 2.76, 0.87
47	29, 62.1	29.3, 62.0	0.0866	0.66, 2.25, 0.83
25	31, 80.7	30.8, 80.8	0.0487	0.48, 2.25, 0.76
11	35, 65.7	34.5, 65.1	0.6836	0.82, 2.49, 0.69
14	37, 75.7	36.3, 75.8	0.5523	0.83, 2.75, 0.62
2	37, 78.4	36.2, 78.6	0.7682	0.81, 2.80, 0.63
46	45, 80.0	44.7, 80.4	0.2510	0.43, 1.90, 0.47
27	46, 87.0	46.1, 87.7	0.4917	0.76, 2.96, 0.46
16	50, 92.0	50.4, 92.3	0.2233	0.50, 2.46, 0.41
43	54, 89.0	53.9, 89.0	0.0214	0.72, 2.88, 0.38
41	55, 87.3	55.2, 86.8	0.2261	0.54, 2.32, 0.36
23	57, 84.2	56.8, 84.4	0.0744	0.79, 2.71, 0.35
9	57, 87.7	57.2, 87.1	0.4089	0.78, 2.92, 0.35
21	65, 90.8	64.8, 91.2	0.1997	0.53, 2.24, 0.29
26	83, 94.0	83.3, 94.2	0.1463	0.47, 2.14, 0.16

The value of SYL (last ACT-R parameter in the genotype) represents seconds per syllable in speaking the solution of each subtraction problem. Using the ACT-R vocal module, the model speaks the subtraction answers as the human subjects do. During the experiment nearly all subjects spoke the subtraction answers out in full. For example, the answer 8185 was spoken as "eight thousand one hundred and eighty five" (about 8 to 9 syllables), instead of "eight one eight five" (4 syllables). An average performing subject would speak between 368 to 414 syllables during one block of serial subtraction. Viewing Table 3 from the bottom up the SYL parameter values show a nearly perfect increase as performance decreases, with the exception of subjects 14 and 2 that are misordered by 0.01. The range for SYL in the individual subject fits is a speedy 0.16 to a slow-speaking 0.87 seconds/syllable, a substantial difference of 0.71. These results show top performers speaking a syllable much more quickly than the poor performers. Surprisingly, all individual subject fits show SYL values greater than the architecture's default value of 0.15.

In Table 3, the BLC value component of the genotypes shows only one value under 2.0; subject 46, one of the average performers, has a BLC of 1.90. This subject also has the lowest ANS value (0.43). BLC is the base level constant of the activation sub-symbolic process affecting both retrieval probability and retrieval time. Overall, Table 3 shows low ANS values associated with low BLC values (subjects 26, 25, and 46), and similarly, high ANS values associated with high BLC values (subjects 1, 2, 9, 14, 23, 27, and 43).

As mentioned previously, the lowest fitness value in Table 3 corresponds to subject 1, the worst performer. The ANS part of the genotype that produced subject 1's lowest fitness value is 0.83. ANS is ACT-R's activation noise parameter. The value 0.83 is higher than what is normally considered reasonable within the ACT-R modeling community. Cognitive modelers using traditional manual optimization would generally not assign a value for ANS that is over 0.5. In Table 3 we see that 60% of the values for ANS are substantially above 0.5 and give good fits.

Table 3 lists only one fit for each subject—the fit resulting in the lowest fitness value. In actuality, each PGA produced a set of good fits less than 1.0 during each run. When all the parameter sets yielding good fits were analyzed, the patterns described above held true. Subsequently, two of the three patterns can be linked to theories of working memory and stress—reported in Kase (2008).

The results described here suggest that an automated and extensive search of the model and architectural parameter space aided by a search algorithm and parallel processing on a cluster computing resource is an ideal validation environment for investigating the affects of moderator influenced behavior (i.e., stress) and individual differences in cognitive performance. The results provide better fits than would be obtained by manual optimization (although not tested, the fits appear as tight as ever reported). The fits also reveal suggestions about the architecture. The PGA will adjust parameters to values within any specified range. Exploring parameter spaces beyond the acceptable modeling norm (e.g., ANS values over 0.5) comes at a negligible cost. Additionally, as a secondary outcome of the fitting process, erroneous architectural default values for parameters can be detected (e.g., SYL default value of 0.15).

7. Comments and Conclusion

The HPC and PGA validation approach presented in the paper integrates parallel processing on high-performance computing clusters with random search

algorithms, such as PGAs, to effectively search without human limitations model and architectural parameter spaces for best fitting sets of parameter values producing performance predictions matching human data. This validation approach could potentially supersede the traditional manual optimization techniques used throughout the cognitive modeling community.

During the testing of the HPC and PGA validation approach with the prototype serial subtraction cognitive model, one validation run of fitting the model to an individual subject's performance data evaluated a total of 20,000 ACT-R parameter combinations in a 3-D parameter space (ANS, BLC, and SYL). Each validation run used only 112 to 176 minutes of runtime on 200 processors. In approximately one day and a half the validation approach could fit the serial subtraction model to all 15 subjects participating in the experiment returning nearly perfect fits for all subjects, showing how they changed.

The HPC and PGA validation approach is a powerful model-fitting methodology that could potentially lead to misuse as warned by Rodgers and Rowe (2001). This new validation approach could either contribute to Roberts and Pashler's (2000) three problems of judging theories by goodness of fit, or enable solutions to these problems (p. 363). Possible HPC and PGA validation enabled solutions include: varying each free parameter over its entire range in all possible combinations; modeling data variability—the distributions of the performance statistics; identifying what the theory cannot fit by manipulating the parameter space or attempting to fit data beyond plausible experimental results—generating simulated behavior; and comparing predictions of competing theories in parallel.

Aside from theoretical testing concerns, the increased accuracy, efficiency, and non-biasness that this validation approach has to offer is available to modelers and extendable in several ways. The approach can be used to fit other cognitive models besides the serial subtraction model. In theory, any parameterized cognitive model could be modified in the same way as our model to run in a parallel processing environment. The primary modifications required are listed in Table 4.

Additionally, different types of search algorithms could be applied, even in combination, to find the best model-to-data fits. For this research, the majority of the search algorithm code was written separately from the cognitive model, interfacing with the model only in the fitness function. For example, the basic code for the PGA was taken from a textbook and then modified to incorporate the running of the ACT-R cognitive architecture and serial subtraction model.

Table 4: Summary of modifications needed to fit a cognitive model using the HPC and PGA validation approach.

1. Specialized front-end function written in the language of the cognitive model (e.g., Lisp) to start up the architecture and model from a call within the search algorithm code.
2. A representation for the model's parameter values (e.g., genetic algorithm genotypes encoded as ACT-R parameter values) that can be passed as arguments through the front-end function to the model.
3. An evaluation of model prediction fitness implemented as a fitness function used by the search algorithm.
4. If a graphical display is required, it should be simulated.

Lastly, for this research, a large cluster computing resource was used for the validation of the serial subtraction model. Most universities have some type of cluster computing resource available to faculty and students. With the exception of the serial subtraction model, all other applications used here are open source (CMUCL, ACT-R) or reside on the cluster itself (C, MPI). With the increased availability of open source applications and academic high performance computing resources, and easy integration of a cognitive architecture and model, this type of validation approach could be adopted for use by cognitive modelers using different architectures.

8. Acknowledgements

This project was partially supported by ONR grant N000140310248. Computational resources were provided by TeraGrid DAC TG-IRI070000T and run on the NCSA clusters. The authors would like to thank Laura Klein and her lab at the Department of Biobehavioral Health, Penn State University, for collection of the human performance data. Discussions with Dave Davis and Michael Schoelles informed our thinking.

9. References

Anderson, J. R. (1993). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Cantu-Puz, E. (2001). *Efficient and accurate parallel genetic algorithms*. Kluwer Academic Publishers.

Glenn, F., Neville, K., Stokes, J., & Ryder, J. (2004). Validation and calibration of human performance models to support simulation-based acquisition. In *Proceedings of the 2004 Winter Simulation*

Conference (pp. 1533-1540).

Gluck, K. & Pew, R. (2001). Lessons learned and future directions for the AMBR model comparison project. In *Proceedings of the Tenth Conference on Computer Generated Forces and Behavioral Representation* (pp. 113-121). Orlando, FL: SISO, Inc.

Gluck, K., Scheutz, M., Gunzelmann, G., Harris, J., & Kershner, J. (2007). Combinatorics meets processing power: Large-scale computational resources for BRIMS. In *Proceedings of the Sixteenth Conference on Behavior Representation in Modeling and Simulation (BRIMS)* (pp. 73-83, 07-BRIMS-037). Orlando, FL: Simulation Interoperability Standards Organization.

Goldberg, D. (1994). Genetic and evolutionary algorithms come of age. *Communications of the ACM 37*(3), 113-119.

Kase, S. E. (2008). *Parallel genetic algorithm optimization of a cognitive model: Investigating group and individual performance on a math stress task*. Unpublished doctoral dissertation, Pennsylvania State University, University Park, PA.

Kirschbaum, C., Pirke, K. M., & Hellhammer, D. H. (1993). The Trier Social Stress Test – A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology 28*, 76-81.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1997). Modeling individual differences in a digit working memory task. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 460-465). Mahwah, NJ: Erlbaum.

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

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

Neville, K., Takamoto, N., French, J., Hursh, S. R., & Schiflett, S. C. (2000). The sleepiness-induced lapsing and cognitive slowing (SILCS) model: Predicting fatigue effects on warfighter performance. In *Proceedings of the 44th Annual Meeting of the Human Factors and Ergonomics Society*, 3-57-3-60. Santa Monica, CA: Human Factors and Ergonomics Society.

Raymond, W. D., Fornberg, B., Buck-Gengler, C. J., Healy, A. F., & Bourne, L. E. (2008). Matlab optimization of an IMPRINT model of human behavior. In *Proceedings of the Seventeenth Conference on Behavior Representation in Modeling and Simulation (BRIMS)* (pp. 26-33). Providence, Rhode Island: Simulation Interoperability Standards Organization.

Ritter, F. E. (1991). Towards fair comparisons of connectionist algorithms through automatically generated parameter sets. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society* (pp. 877-881). Hillsdale, NJ: Erlbaum.

Ritter, F. E., Reifers, A. L., Schoelles, M., & Klein, L. C. (2007). Lessons from defining theories of stress for architectures. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 254-262). New York, NY: Oxford University Press.

Ritter, F. E., Schoelles, M., Klein, L., & Kase, S. E. (2007). Modeling the range of performance on the serial subtraction task. In *Proceedings of the 8th International Conference on Cognitive Modeling (ICCM)* (pp. 255-260). Ann Arbor, Michigan. UK: Taylor & Francis/Psychology Press.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review 107*(2), 358-367.

Rodgers, J., & Rowe, D. (2002). Theory development should begin (but not end) with good empirical fits: A comment on Roberts and Pashler (2000). *Psychological Review 109*(3), 599-604.

Sargent, R. G. (2005). Verification and validation of simulation models. In *Proceedings of the 2005 Winter Simulation Conference* (pp. 130-143).

Tor, K., & Ritter, F. E. (2004). Using a genetic algorithm to optimize the fit of cognitive models. In *Proceedings of the Sixth International Conference on Cognitive Modeling* (pp. 308-313). Mahwah, NJ: Erlbaum.

Author Biography

SUE E. KASE is a Postdoctoral Fellow at the Defense Threat Reduction Agency (DTRA), Directorate of Basic and Applied Science, in Fort Belvoir Virginia. The fellowship is a collaboration with Penn State University, College of Information Sciences and Technology, through the University Strategic Partnership Program.

FRANK RITTER is on the faculty of the College of Information Sciences and Technology, an interdisciplinary academic unit at Penn State University to study how people process information using technology. He edits the *Oxford Series on Cognitive Models and Architectures* and is an editorial board member of *Human Factors, AISBQ,* and the *Journal of Educational Psychology*.