Operator Centered Design of Ship Systems

Susan F. Chipman, Ph.D., U. S. Office of Naval Research

David E. Kieras, Ph.D., University of Michigan

UNCLASSIFIED/DISTRIBUTION UNLIMITED

Abstract

Cognitive science research, much of it supported by the Office of Naval Research, is bringing about a scientific revolution in our understanding of the human operator. It is yielding computational theories of human cognition and perceptual/motor activity that provide precise quantitative predictions of important variables such as the times required to complete tasks or to learn them in training. Although the scope of coverage of these theories is limited and basic research aimed at expanding them is on-going, they already have much to offer in aiding the design of ship systems that will optimize the combined effectiveness of human operators and the systems they will be using. This presentation discusses 1) what can be done now, 2) tools under development that will facilitate the use of these theories, reducing the labor involved, and 3) a long-term vision for what might be achieved in this area.

Appearing in Proceedings of “Engineering the Total Ship Symposium, 2004” Held at NIST, Gaithersburg, MD March 17-18, 2004, organized by the American Society of Naval Engineers.

The Navy is a high technology enterprise that makes extreme demands on the human operators who operate its systems, both in performing their jobs and in learning to perform them. Given current goals to reduce manning levels on the Navy's ships, these demands are becoming still more extreme. For that reason, it is important that future ship systems be as well designed as possible to minimize the demands made on the human operator. As a reminder of the problem we face, Figure 1 is a picture of an actual workstation aboard an AEGIS ship.

Figure 1: *Workstation on an AEGIS ship.*

Computational Theories of Human Cognitive Architecture

The goal of this paper is to explain how ONR's long range basic research investments in computational theories of human cognitive architecture are now beginning to pay off by providing a sound scientific basis for operator centered design of ship systems. The term *cognitive architecture* is an analogy to the notion of computer architecture and refers to the (relatively) fixed features of the human information processing system, the basic characteristics of the typical human operator of Navy systems. These theories can be viewed as special kinds of computer "languages" in which human skills are written. They are special languages that embody the constraints on what people can do, reflecting what we have learned in 100 years of psychological research. ONR has been involved in supporting several different variations on this theoretical theme. The different theories are differentiated primarily by the particular problem areas or types of human performance that the researchers have chosen to focus on. All are efforts at theory unification, bringing together numerous factoids from past psychological literature in a way that makes it possible to determine their joint implications for practical applications.

The leading cognitive architectural theory at this time is that of John R. Anderson of Carnegie-Mellon University, known as ACT-R[1] (Anderson, 1993; act-r.psy.cmu.edu). Anderson's research and theorizing has emphasized human problem solving and the learning of complex cognitive skills, with applications to artificially intelligent training systems. For example, a tutor of high school algebra based on his work is now extensively used in schools throughout the country, including the DoD Dependent School System. A substantial national and international research community is now working in Anderson's theoretical framework modeling many different kinds of human cognitive performance, including some applications to human computer interaction. Two other researchers at Carnegie-Mellon University, Carpenter and Just, developed a cognitive architecture known as CAPS (Just, Carpenter & Varma, 1999) that emphasizes modeling the capacity limitations on human information processing, as well as individual differences in those capacities. More recently, their research program has moved into modeling of the associated brain activity as well, which is revealed by functional magnetic resonance imaging (fMRI) during task performance. The imaging of brain activity provides an independent, triangulating source of evidence that strengthens these cognitive theories. David Meyer and David Kieras at the University of Michigan have a cognitive architectural theory known as EPIC that has emphasized developing an explanation for the complex results of experiments in which people are asked to do more than one task at a time. EPIC was also the first theory of this type to integrate models of human perception and motor control, and Kieras has been particularly interested in applications to human computer interaction. The first major accomplishment of this research effort was to show that the phenomena of interference between multiple tasks that had been taken as evidence for limitations on the human's central executive processing capacity could actually be accounted for by competition for where the eyes are looking, where the hands are moving, and what strategies people adopt to coordinate the tasks (Kieras & Meyer, 1997). People can do more than we have thought if task inputs and outputs are designed to avoid these conflicts, and they adopt strategies that take advantage of these possibilities. Gentner and Forbus of Northwestern University have a theory known as MAC/FAC (for "many are called but few are chosen") that emphasizes analogical reasoning and the retrieval of past experience from memory (Forbus, Gentner & Law, 1995). This theory has strong potential for applications in case-based instruction and decision aiding. In addition, ONR has supported a number of researchers modeling human cognition in Allen Newell’s cognitive architecture, SOAR, the development of which was supported primarily by DARPA. Notably, Bonnie John of Carnegie-Mellon has used SOAR to study how humans interact with computer displays. However, SOAR is really an AI programming system inspired by human psychology, and is unrealistic as a model of human psychology in several important ways. John is now working in the ACT-R theoretical framework.

All of these cognitive architectural theories can be seen as part of a larger movement toward a unified theory of cognition, perception and action (Newell, 1991). Psychology has been seen as a soft science, although there have been many “hard” bits, scientific laws of small scope, for a long time. In the development of these theories, we are seeing a scientific revolution in progress that is making it possible to make precise quantitative predictions about human performance in quite complex situations.

The EPIC Cognitive Architecture

Figure 2 is a cartoon representing the structure of the EPIC cognitive architecture relating to a task environment. It contains separate perceptual, cognitive and motor systems operating in parallel. In fact, there are 3 motor systems shown as operating in parallel: the oculomotor processor moving the eyes, the vocal motor processor generating speech and a manual motor processor controlling the two hands. These features reflect research evidence of independence and interference. The two hands do not operate independently of each other. Note that the model has no feet. It is still a

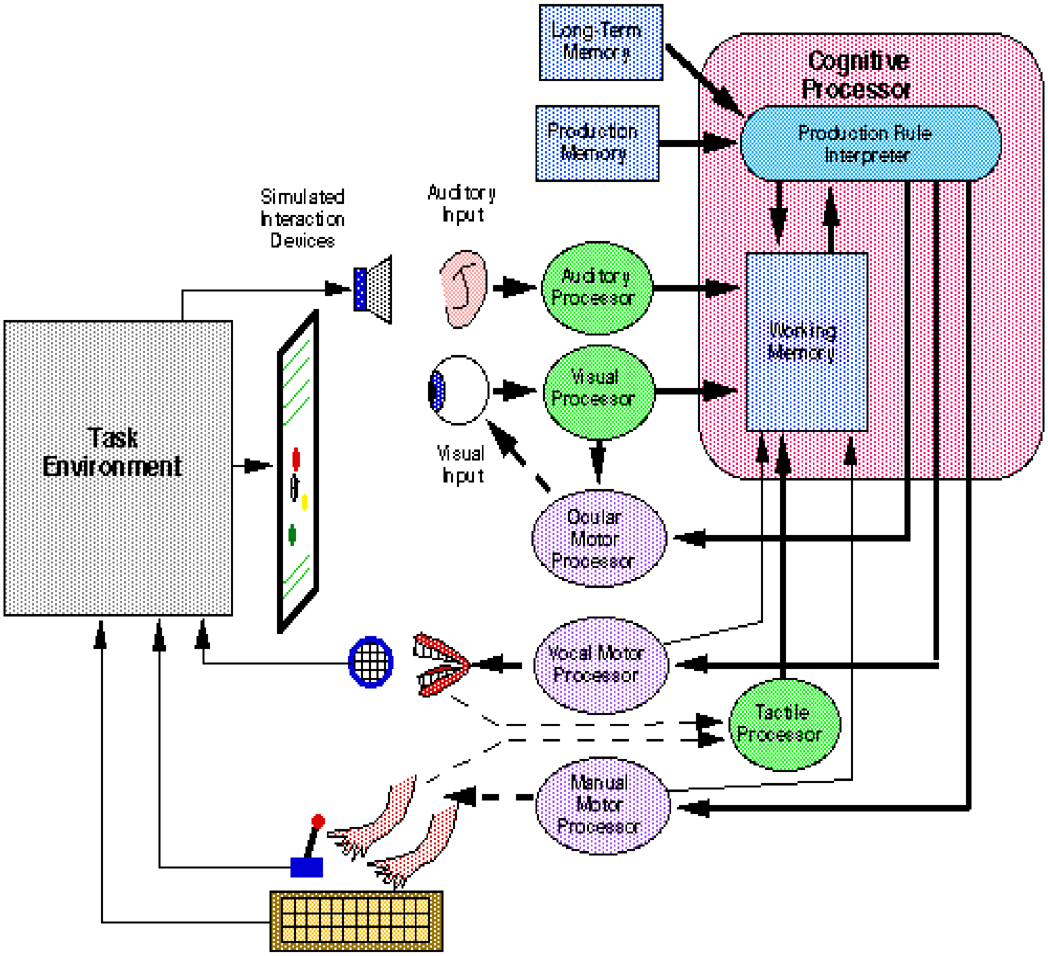


Figure 2: *Structure of the EPIC Cognitive Architecture*

work in progress, and no tasks have yet been modeled that involve the use of the feet. Similarly there are separate auditory, tactile, and visual processors. The model of visual processing is quite complex, reflecting what we know about what kinds of information can be picked up at what positions in the visual field of the eye, and the fact that eye movements may be required to access task information. Task strategies are represented in EPIC, and in the other cognitive architectures mentioned above, as *production rules.* Production rules have an “if-then” structure. If the necessary conditions are present, either in the external world, such as a display, or in the internal mental contents of what is known as *working memory*, these rules execute. The actions of these rules either control motor actions or modify the contents of working memory. These rules can be viewed as continuously looking for the appropriate conditions, so a computational model built from production rules is not like an ordinary computer program with a fixed sequence of operations, but a program which can respond in very dynamic ways to the current situation. The central executive cognitive processor of EPIC has a working memory of immediately available information as well as a production rule interpreter. It is supported by long term memory for factual (declarative) information and by a memory for production rules that have been learned. The cognitive processor allows for parallel processing, overlapping the processing of different aspects of the task or tasks in time.

Addressing the Problems Posed by Novel Systems

Having these cognitive architecture "languages" enables us to address important military applications in the area of human computer interaction that are beyond the capability of past HCI development methods. These are the challenge of designing entirely new systems – as opposed to refining or improving existing systems, and the challenge of developing training for systems that do not yet exist. Ideally one would like to have training available at the same time as the system is completed. If you can write the program for an operator skill in a cognitive architectural language, you can perform evaluative analyses of designs before prototypes are even built. You can also estimate the operator's cognitive workload, essentially as the density of cognitive operations going on at a moment of time (Lebiere, 2001). If it gets too high at any point, errors are likely to occur, and redesign of the task is called for. Based on this skill program, you can also develop an artificially intelligent tutor to train the operator, without having to study experienced human operators who do not yet exist. If the skill can be written in the architectural language, then the human operator should be able to learn it, although the resulting skill might be slightly different from what would have been developed naturally, without the training. A key part of the approach is writing the skill descriptions by using a modeling approach that represents the full scope of the skill in its general form, rather than just the activity required in a specific scenario. This makes the complete skill description available for training purposes, but it also means that the model can be used to predict performance in a wide variety of different specific situations. This makes it easy to evaluate the system design comprehensively.

Relating to Drew's paper in this same workshop, these languages could also be used to program intelligent agents, although this has not been the practice to date. (SOAR, originally designed as a programming approach for artificial intelligence rather than as a veridical model of human cognition, is of course the exception.) However, some of the more advanced artificially intelligent training systems do incorporate task experts written in such modeling languages, and there have been experiments in which such models have served as artificial team members interacting with a trainee. Such models developed for training purposes might also be considered a step towards automation of the functions performed by human team members. The standards that must be met by such artificial experts for training system use are not as high as for turning decisive responsibility over to artificial experts. In a training scenario, "truth" is known, and one can test the system against the training scenarios to ensure that its behavior is reasonable. There might be some significant advantages to using cognitive architectural languages for writing intelligent agents or expert systems. One of the earliest lessons learned from work on expert systems for medical diagnosis is that their lines of reasoning must be comprehensible to humans if the systems are to gain acceptance.

An Engineering Model Approach to Ensuring System Usability

Today it is possible to use model predictions instead of user testing for a large fraction of the system design effort (Kieras, 1988, 1997, 1999, 2003). In order to do this, one must describe the interface design in detail and build a model of the user doing the task. The model can then be used to predict the time required to execute the task(s), the time required to learn the task(s) – based on the number and complexity of production rules that must be learned, and to estimate cognitive workload during task performance. Based on these predictions, one can revise the interface design or chose from among several proposed alternative designs. One gets usability results *before* building a prototype of the system. Thus, the engineering model approach makes it economically feasible to go through more design iterations with successive refinements. It summarizes the interface as seen from the operator's point of view. We are not yet able to model all aspects of the operator's interaction with the system, however, so empirical user testing remains an important final check on the analysis, as well as insurance that mistakes have not been made in building the model.

Why This is Worth Doing – The Tomahawk Story

A long time ago, Glenn Osga of SPAWAR Systems Center did a GOMS analysis of a proposed new launch system for the Tomahawk missile. **GOMS** stand for **G**oals, **O**perators, **M**ethods (sequences of operators), and **S**election rules for choosing methods.

GOMS (Card, Moran & Newell, 1983) is the earliest form of the type of analysis and user modeling being advocated here, and includes a range of specific techniques, ranging from hand or spreadsheet calculations (such as that done by Osga) to full simulations of the human-computer interaction (John & Kieras, 1996 a&b). On the basis of the analysis, Osga predicted that the launch process would take too much time (task execution time) to meet the requirement. The customer did not like the message[3]. The study was locked deep in a safe, and the system was built as designed. Years later, the system failed its acceptance test. This story was quite accidentally revealed in a moment of drama at workshop with diverse participants brought together by the DD-21 program. Osga was speaking and mentioned having once done this analysis. In the audience was the officer assigned to investigate why the launch system had failed its acceptance test. Not surprisingly, the final report on his investigation recommended that GOMS analyses be done on all proposed new systems. Undoubtedly, it cost a great deal of money to ignore that original prediction and built the system, only to have it fail the acceptance test. . Today, fortunately, Osga is again working on the Tomahawk system interface.

An Instructive Example of Cognitive Modeling: The NRL Automation Deficit Task

Many of the ideas about how to use cognitive modeling for advance design of usable systems were developed through work modeling human performance on a simplified laboratory task developed at the Naval Research Lab by James Ballas to study the phenomenon of “automation deficit.” In the DARPA Pilot’s Associate program, automation deficit emerged as a significant problem. When some function has been handled by an automated system for a period of time and then the human pilot is asked to take it over, the human pilot’s performance is initially inferior to what would ordinarily be expected. This is automation deficit. Ballas developed a simplified analog of a fighter cockpit task to study this phenomenon. It combines target classification and target tracking, and it did succeed in demonstrating automation deficit when the target classification task was automated. The task interface is illustrated in Figure 3.

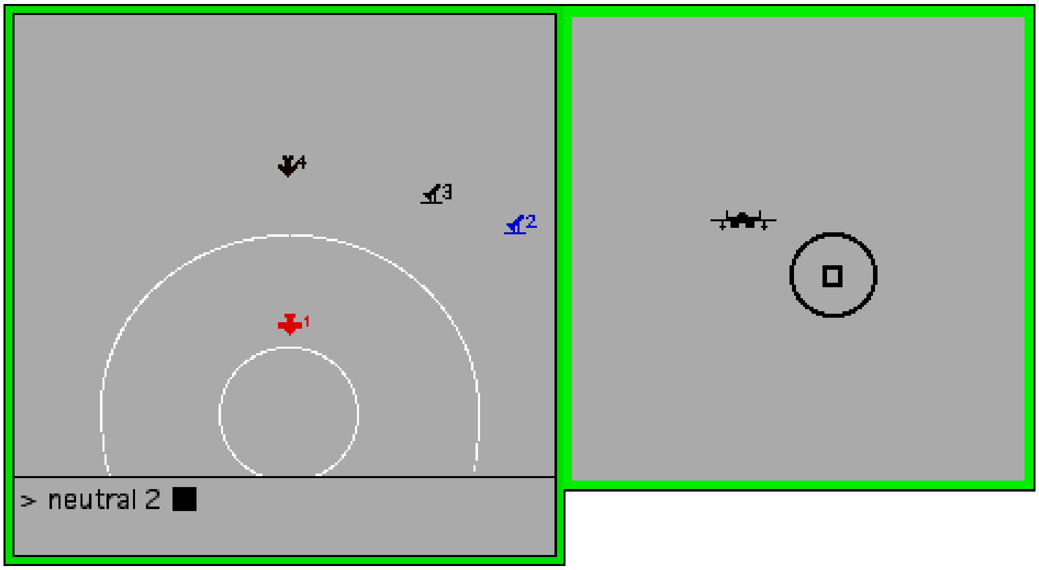


Figure 3: *Interface of the NRL task.*

The Process of Building a Cognitive Model

There are three major elements that go into the building of a cognitive model of task performance. One is the (relative) constant of the human cognitive architecture. Secondly, one must identify the mandatory requirements of the task, what the operator absolutely *must* do to perform the task. The third, and most difficult element, is specifying the strategy that the operator will use to meet those requirements. Usually there are several, or even many, options for the operator's task strategy. In modeling human performance in the NRL task, Kieras and Meyer found that matching the speed of actual human performance in the task required a great deal of overlapped parallel processing. These have not been typical assumptions about human cognitive processing, and they were not the first hypotheses considered, by any means. Ultimately the researchers realized that human operators also changed strategies depending upon task workload. When the workload was light, there was less overlapping and parallel processing. Not surprisingly, humans do not choose to perform at maximal possible speed if the task does not require it.

The cognitive modeling did provide a very simple, comprehensible explanation for the automation deficit phenomenon. When people are responsible for performing both components of the NRL task and are executing the tracking task, they nevertheless notice the appearance of targets on the target screen and form a mental queue of the targets in the order and location of their appearance. So, when they switch to execution of the target classification task, they know what to process, in what order. However, when the target classification task has been automated, they do not attend to the appearance of the targets and do not form that mental queue. So, when they are asked to resume responsibility for the target classification, people must catch up on the situation, perhaps doing things out of order, leading to a period of relatively poor performance.

Many lessons were learned from the modeling of performance in the NRL task, but for system design, the critical question is what to do if you don’t have human performance data to model.

The Bracketing Approach to the Problem of Unknown Strategies

Kieras and Meyer (2000) proposed a bracketing approach to the problem of unknown strategies that the human operator will use to perform a task in a proposed future system design. One can determine the fastest possible strategy, given the constraints of the human perceptual/cognitive/motor architecture: often eyes or even the head must move to see vital information (Marshall, 2002; Marshall, Morrison, Allred, Gillikin, & McAllister, 1997; Morrison, Marshall, Kelly & Moore, 1997). These movements take substantial, approximately known amounts of time. A hand cannot type and move a mouse or trackball at the same time, so these actions cannot go on in parallel, and so on.   One can also determine the slowest reasonable strategy that meets the mandatory requirements of the task but does not involve much overlapping or parallel processing. It can be assumed that actual operator performance will lie in between these two extremes. As Figure 4 shows, the bracketing approach did work for the NRL task data.   In this case, as in a number of others that have been modeled, actual human performance seemed to lie somewhat closer to the fastest possible strategy. True prediction also worked for another version of the NRL task with a touch screen interface, where the human performance data had not been previously modeled (Kieras, Meyer, & Ballas, 2001).

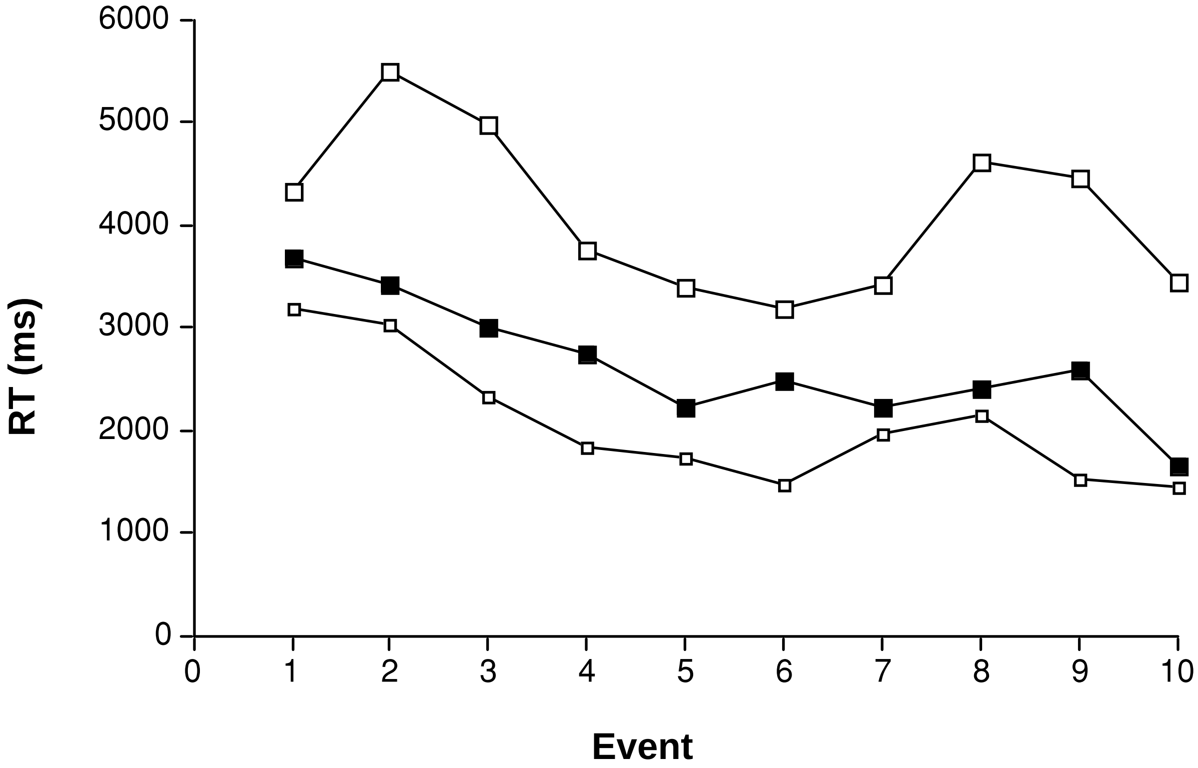


Figure 4: *Actual human performance times in the NRL task (filled squares) are bracketed by the fastest possible times (small open squares) and the slowest reasonable times (large open squares).*

If the fastest possible strategy isn't fast enough to meet the requirements, then you have a serious design problem calling for serious redesign efforts. This appears to have been the situation with the Tomahawk launch system. In most cases, furthermore, you do not want to design so that only the fastest possible strategy will suffice. This could get you into the territory where the relatively small individual differences in processing speed could become critical. Furthermore, people usually will not choose to work at maximum possible speed, especially if they must perform the task over long periods of time. The ideal system would have functionality and an interface that are good enough to permit even the slowest reasonable strategy to be fast enough. This would give the operator considerable reserve capacity to deal with unexpected problems or high stress.

At this point, only the full-fledged cognitive architectural models such as EPIC can easily represent the bracketing strategies. This is a bit of a problem because simpler approaches like GOMS are currently easier to work with. Both types of approach, however, permit exploration of system redesign with much less effort than traditional usability evaluation methods.

The Payoffs of Detailed Analysis and Modeling

We have been emphasizing the capability and value of quantitative prediction of execution times, cognitive workload and learning times. The Tomahawk story highlighted the importance of predicting task execution times, and both execution times and learning times often have important economic implications. However, the qualitative evaluation that detailed GOMS analysis or cognitive modeling enables is at least as important. This can pick up the problems that bother us with most of the computer systems we all must use and that can lead to catastrophic performance failures in the worst cases. Given a GOMS analysis, one can ask these questions:

1) Is the system **complete?** Is there a method available for every goal that the user will pursue?
2) Is the system **natural?** Will the goals make sense to a new user?
3) Is the system **cleanly designed?** Is each method in the system actually useful, with a clear rule for selecting it to use at the right time? (Most readers will be familiar with the tendency for Microsoft products to have many ways of doing the same thing. Having just one good way makes a system easier to learn.)

4) Is the system **consistent?** Do similar goals have similar or the same method for achieving them?
5) Is the system **efficient?** Do frequent goals have short, fast methods? You don't want to have the most frequently used functions buried in the fourth level of a menu, for example.
6) Is there a good method for **error recovery?** Error is frequent, and system designers often overlook the need to provide for operators to recover from error, such as "undoing" the last action. All too often, users find themselves in completely unknown and bewildering territory when they make a small error.

Returning to the subject of quantitative prediction, we will give an example of what was actually achieved in the redesign of a commercial CAD system for biomechanical safety analysis of factory jobs, using GOMS analysis. The GOMS analyses predicted that the redesign would achieve a 40% reduction in the time required to execute a set of typical tasks and a 46% reduction in the time to learn the system. Actual human data showed a 39% reduction in execution time and a 46% reduction in learning time (Gong & Kieras, 1994). Obviously these large improvements have economic significance. Such accurate predictions have occurred in other applications of GOMS analysis.

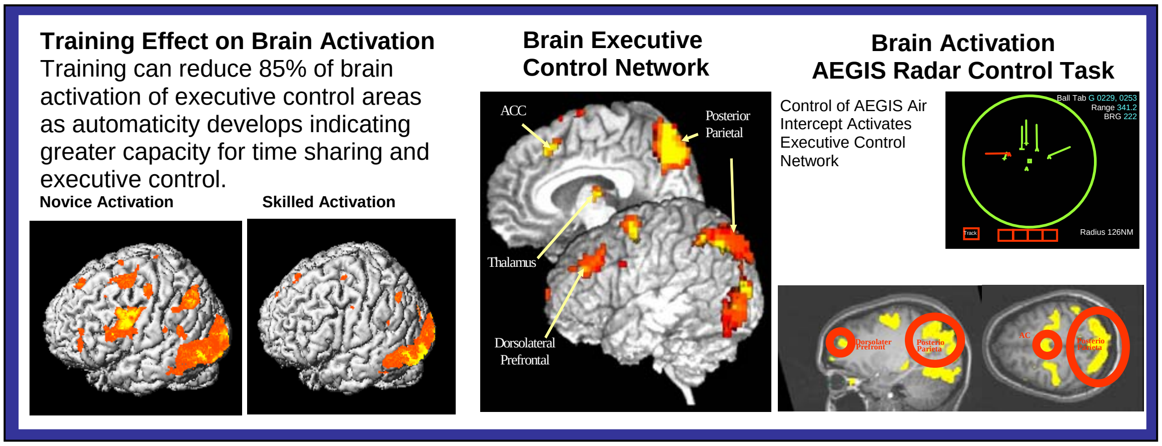
Often, unfortunately, it seems that those who are paying for the design of systems do not want to invest the resources required for a detailed human factors analysis. They often believe that user interface design should be easy, cheap and non-technical. They hope that somebody else, somewhere, will invest a small amount of money in a psychological research project and guidelines will be produced that will take care of all these problems for all time. This is not the case. First, when confronted with the myriad design decisions involved in developing complex software, traditional guidelines for usable design have proved to be vague and conflicting. A more rigorous design discipline is needed. Second, like the design of a bridge or design of the ship itself, design of the systems that the operators will use requires getting the details right, and this requires a significant design effort using the most advanced science available to address the problem. Nobody would suggest that a fire-control radar system could be successfully designed in a few days by non-technical people following informal guidelines based on sketchy or obsolete theory, but that is often how interface design is approached. Reasonable funding must be allocated to operator centered design of the ship's systems if the ship is to achieve its performance goals. For its part, the R&D community is working on developing tools that will reduce the costs of doing this work (eg. Byrne et al., 1994; John et al., 2004).

A Vision for the Future of Operator Center Design of Systems

Today cognitive modeling approaches enable us to compare alternative designs and predict the combined performance of the total system that combines the human operator with the non-human components. However, this capability is not being used by the Department of Defense on any significant scale[2]. In the future, we hope that the predictive evaluation of systems will become routine. Tools are now being developed that will significantly reduce the labor involved in doing these model-based evaluations. For example, Williams' CAT-HCI (Williams, 2000) is a program that aids and records GOMS style task analysis, automatically computing execution times and estimated learning times. It incorporates the information from the psychological research literature about the time required for various component actions and cognitive processes. Kieras' GLEAN tool for GOMS analysis (Kieras, 1999) uses a simplified cognitive architecture to allow powerful human performance simulation models, including models of teams of operators (Kieras & Santoro, 2004; Santoro, Kieras, & Pharmer, in press), to be constructed much more easily (though with lower fidelity) than with the full-fledged cognitive architectures such as EPIC or ACT-R. Most recently, some experimental tools developed by Bonnie John and her colleagues (John, Prevas, Salvucci, & Koedinger, 2004) enabled extremely fast model development for certain simple cases. The next few years are likely to see rapid development and improvement of such modeling tools.

Continuing basic research on the computational theories of human cognitive architecture will be bringing us more strongly validated and more complete models of human perceptual/cognitive/motor architecture. For example, today we cannot model the complex three-dimensional reasoning and problem solving such as that involved in the task of a submarine attack officer, although we can model his interaction with the system that provides information supporting this task. Two younger researchers are now working on extending the ACT-R cognitive architecture to handle such spatial thinking and reasoning. Today we are best at modeling the relatively simple forms of perception and motor action that are typically involved in interacting with computer displays and computer input devices, as opposed to the full complexity of perception and action in the natural world. Fortunately, however, that capability covers a great deal of the territory of interest in the design of military systems. A few are also beginning to work on representing how stress, fatigue, and other such variables of a physiological character affect human performance within the framework of cognitive architectural theories. This is obviously important for the design of military systems, whose operators may not be functioning at their best. We believe that individual differences in cognitive capacities can probably be represented as variations in some of the key parameters in these cognitive architectural theories, such as the size of working memory. This remains to be seen through future research. Although individual differences are always of high interest to people, what we almost all have in common as humans is most important to system design. The differences are small variations on the common theme.

In the hands of clever experimental psychologists who know how to design the critical revealing experiments, relatively new and rapidly advancing brain imaging technology is beginning to provide triangulating evidence for the cognitive architectural theories that have been based on other kinds of evidence. For example, recent research under a large Multidisciplinary University Research Initiative grant to Carnegie Mellon and the University of Pittsburgh has provided evidence identifying the “executive control system of the human brain” that is active when people are learning new skills or dealing with unfamiliar problems. When skills are highly practiced, “overlearned”, they become much less subject to disruption and the associated activity in these brain areas is greatly reduced. See Figure 5, from a poster presentation of recent results from this research. In the near future, we can expect to be learning much more about how activity in these regions of the brain is related to perceived cognitive workload and performance in multitasking situations. Similarly, such imaging allows us to see whether different tasks, or different components of a complex task, are calling upon the same processing resources in the brain and are therefore likely to conflict with each other, or whether they are calling upon different processing resources and are likely to proceed successfully in parallel.



Training Effect on Brain Activation
Training can reduce 85% of brain activation of executive control areas as automaticity develops indicating greater capacity for time sharing and executive control.

Brain Executive Control Network

Brain Activation AEGIS Radar Control Task

Control of AEGIS Air Intercept Activates Executive Control Network

Figure 5: *Recent results from Cognitive, Biological & Computational Analyses of Automaticity in Complex Cognition: MURI-ONR Grant N000140110677 PI Marcel Just (Carnegie Mellon University), Cleotilde Gonzalez (Carnegie Mellon University) & Walter Schneider (University of Pittsburgh) . The same reductions in control system activity were found in a simulated AEGIS task as in simpler laboratory tasks. Background: Schneider, W. & Chein, J. M. (2003)*

For the specific problem of cognitive engineering of operator centered systems, one gleaming prospect for the future is the possibility of automating the design process. One can view the human cognitive architecture as a very complex function that converts prospective system designs into performance figures of merit that interest us. Perhaps we could invert that complex function and automatically generate an optimal design. This would certainly be difficult. It might be impossible. But Barnsley (1994) did succeed in inverting fractals and yielding impressive image compression techniques. Maybe this vision of the future is possible too.

For those who would like to pursue these applications, in addition to the references cited in this paper, there are training opportunities. An ACT-R Summer School is held annually at Carnegie Mellon University and there are also training materials that can be downloaded for self study: http://act-r.psy.cmu.edu/workshops/workshop-2004/summerschool_2004_announce.htm. Kieras has proposed a workshop for the International Conference on Cognitive Modeling that will take place July 29-August 1, 2004, in Pittsburgh: http://simon.lrdc.pitt.edu/~iccm/. Similar workshops are likely to be presented at the annual CHI conference. ONR plans to sponsor a virtual workshop by Kieras sometime in the next fiscal year that would be available over the Internet. Persons wanting to be notified of that workshop when it happens should send an email message to Susan Chipman, chipmas@onr.navy.mil.

Notes

Note: Most of the specific ideas about operator centered design in this paper are from the work of David Kieras, although this particular presentation of them was primarily prepared and written by Susan Chipman. Susan Chipman manages the Cognitive Science program at the Office of Naval Research, which has supported much of that work.

1. The names of the cognitive architectures are better thought of as names than as acronyms. The translation of ACT has changed over time. One translation is “adaptive control of thought.” SOAR was once an acronym but is now a name.

EPIC stands for “Executive-Process/Interactive Control.” CAPS or 3CAPS is “Capacity-Constrained Collaborative Activation–based Production System.”

2. There is some significant use of MicroSaint modeling that reflects an older generation of cognitive theory without the same inherent predictive capability that models based on theories of cognitive architecture have.
3. Osga now believes that the negative reaction was largely due to the fact that the system design process had already gone too far by the time the analysis was done; change at that point would have been quite costly. It is important to do these analyses from the very beginning of a project.

References

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

Barnsley, M. F. (1994) *The Fractal Transform.* London: AK Peters Ltd.

Byrne, M. D., Wood, S.D., Sukaviriya, P., Foley, J.D., & Kieras, D. E. (1994) Automating Interface Evaluation. In *Proceedings of CHI, 1994,* Boston, MA, USA, April 24-28, 1994. New York: ACM, pp. 232-237.

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

Forbus, K. D., Gentner, D., Law, K. (1995) MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19, pp. 141-205

Gong, R., & Kieras, D. (1994) A validation of the GOMS model methodology in the development of a specialized, commercial software application. In *Proceedings of CHI, 1994*, Boston, MA, USA, April 24-28, 1994. New York: ACM, pp. 351-357.

John, B. E., & Kieras, D. E. (1996a) Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Computer-Human Interaction, 3,* 320-351.

John, B. E., & Kieras, D. E. (1996b) The GOMS family of user interface design and evaluation: Which technique? ? *ACM Transactions on Computer-Human Interaction, 3,* 320-351.

John, B.E., Prevas, K., Salvucci, D.D., & Koedinger, K. (2004) Predictive human performance modeling made easy. Accepted for CHI 2004, April 24-29, 2004, Vienna, Austria.

Just, M. A., Carpenter, P. A.,& Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping, 8,* 128-136.

Kieras, D. E. (1988) Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed) *Handbook of Human-Computer Interaction* (pp. 135-158). Amsterdam: North-Holland Elsevier.

Kieras, D. E. (1997) A guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. Landauer, & P. Prabhu (Eds),) *Handbook of Human-Computer Interaction* (Second Edition) (pp. 733-766). Amsterdam: North-Holland Elsevier.

Kieras, D.E. (1999) A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN3. Document and software available via anonymous ftp at ftp://www.eecs.umich.edu/people/kieras

Kieras, D. E. (2003) Model-based evaluation. In J. Jacko & A. Sears (Eds) *The Human-Computer Interaction Handbook.* Mahwah, NJ: Lawrence Erlbaum Associates. 1139-1151.

Kieras, D. E. & 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. & Meyer, D. E. (2000) The role of cognitive task analysis in the application of predictive models of human performance. In J. M. C. Schraagen, S. F. Chipman, & V. L. Shalin (Eds) Cognitive Task Analysis. Mahwah, NJ: Lawrence Erlbaum Associates, 237-260.

Kieras, D., Meyer, D, and Ballas, J. (2001) Towards demystification of direct manipulation: Cognitive modeling charts the gulf of execution. In *Proceedings of CHI, 2001*, Seattle, WA, , USA, March 31- April 4, 2001, New York: ACM, pp. 128-135.

Kieras, D. E. & Santoro, T. P. (2004) Computational GOMS Modeling of a Complex Team Task: Lessons Learned. In: *CHI 2004*, April 24-29, 2004, Vienna, Austria.

Lebiere, C. (2001). A theory-based model of cognitive workload and its applications. In *Proceedings of the 2001 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC).* Arlington, VA: NDIA

Marshall, S. P. (2002). The Index of Cognitive Activity: Measuring cognitive workload. In *Proceedings of the 2002 IEEE 7th Conference on Human Factors and Power Plants* (pp. 7.5 – 7.9). New York: IEEE.

Marshall, S. P., Morrison, J. G., Allred, L. E., Gillikin, S., & McAllister, J. A. (1997). Eye tracking in tactical decision-making environments: Implementation and analysis. In *Proceedings* *of the 1997 Command and Control Research and Technology Symposium.* Washington, DC: National Defense University.

Morrison, J. G., Marshall, S. P., Kelly, R. T., & Moore, R. A. (1997). Eye tracking in tactical decision-making environments: Implications for decision support. In *Proceedings of the 1997 Command and Control Research and Technology Symposium.* Washington, DC: National Defense University.

Newell, A. (1991) Unified Theories of Cognition. Cambridge, MA: Cambridge University Press.

Santoro, T. P., Kieras, D. E., & Pharmer, J. (in press) Verification and validation of latency and workload predictions for a team of humans by a team of GOMS models. *U.S. Navy Journal of Underwater Acoustics, Special Issue on Modeling and Simulation.*

Schneider, W. & Chein, J. M. (2003) Controlled & automatic processing: Behavior, theory, and biological mechanisms. *Cognitive Science*, 27, 525-559.

Williams, K. E. (2000) An automated aid for modeling human-computer interaction. In Schraagen, Chipman & Shalin (Eds*) Cognitive Task Analysis.* Mahwah, NJ: Erlbaum., pp 165-180.

Author Information

Dr. Susan Chipman manages the Cognitive Science program at the U.S. Office of Naval Research, as well as more applied programs in advanced training technology. Previously, she was Assistant Director of the National Institute of Education, where she was responsible for managing research programs in mathematics education, cognitive development, computers and education, and social influences on learning and development. Prior to becoming a research manager, her personal research focused on visual pattern perception and its development, and she has also written extensively on the participation of women and minorities in mathematics, science and technology. Books which she has edited are *Thinking and Learning Skills*, *Women and Mathematics: Balancing the Equation*, *Knowledge Acquisition*, *Cognitively Diagnostic Assessment* and *Cognitive Task Analysis*. She received an AB in Mathematics, MBA, and AM and PhD in Experimental Psychology from Harvard University and is a Fellow of both the American Psychological Association and the American Psychological Society.

Address: Code 342, Office of Naval Research, 800 N. Quincy St., Arlington, VA 22217-5660
Telephone: 703-696-4318
Email: chipmas@onr.navy.mil

Dr. David Kieras is a Professor in the Electrical Engineering & Computer Science Department at the University of Michigan, with a joint appointment in the Department of Psychology. He received a BA in Psychology from Rice University in 1969 and a PhD in Experimental Psychology from the University of Michigan in 1974.He has served on the faculty of the University of Arizona as well as the University of Michigan. His primary general research field is applied and theoretical cognitive psychology, with specific interests in human-computer interaction, cognitive simulation modeling, human performance, and natural language processing. His research has been supported by ONR, NASA, IBM, and ARPA. His primary hobby is the Naval history and the history of Naval technology.

Address: Artificial Intelligence Laboratory, Electrical Engineering and Computer Science Department, University of Michigan, 1101 Beal Avenue, Ann Arbor, MI 48109-2110
Phone: (734) 763-6739
Email: kieras@eecs.umich.edu