Using Cognitive Engineering to Improve Systems Engineering

Craig Bonaceto
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
cbonaceto@mitre.org

Kevin Burns
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
kburns@mitre.org

Abstract. Enterprise Systems Engineering (ESE) must go beyond the hardware and software of systems to address Human-Systems Integration (HSI). Towards that end, we propose that Cognitive Engineering techniques can and should play a key role in Systems Engineering efforts. In this paper we survey various methods in Cognitive Engineering, showing where these methods apply to specific problems in Systems Engineering from Concept Definition and Requirements Analysis, through Function Allocation and Performance Estimation, to Training Development and Performance Assurance. We also describe several uses of selected methods, including Cognitive Task Analysis, Computational Cognitive Modeling, and Critical Incident Analysis, to tackle specific problems in air traffic control. Taken together, these specific cases along with our general survey offer a roadmap for using Cognitive Engineering to improve Systems Engineering.

1. Introduction

1.1 Enterprise Systems Engineering

Systems Engineering has traditionally focused on the technological aspects of system design, such as hardware, software and automation, while largely ignoring the fact that these systems will ultimately be used in the service of humans to meet the demands of work domains. This failure to consider humans as key components of an *Enterprise* is a serious issue in the current practice of *Systems Engineering*.

As an example, consider the performance of the U.S. Patriot Missile batteries deployed in the Iraq war (2003), which were set to function with a high degree of automation. Here, "The operating protocol was largely automatic and operators were trained to trust the system's software... a design that would be needed for heavy missile attacks" (Defense Sciences Board, 2005). But the batteries were operating in an environment with few missiles and many friendly aircraft. Moreover, the operators were not adequately trained to recognize that Patriot radar system can be prone to making spurious hits and sometimes identify friendly aircraft as enemy missiles, nor did their displays indicate the inherent uncertainty in target ID. Thus, operators were reasonably predisposed to trust the system's assessments and its decisions to launch missiles against possibly hostile targets. This was a contributing factor in the shoot down of a British Tornado and a U.S. Navy F/A-18, for which a Defense Sciences Board report concluded that "more operator involvement and control in the function of a Patriot battery" will be necessary to overcome the system's limitations (Defense Science Board, 2005).

When confronted with such system failures, the blame is often assigned to system operators – be they pilots, air traffic controllers, nuclear plant technicians, or military personnel. As author James Chiles observes in *Inviting Disaster*: *Lessons from the Edge of Technology*:

> "Too often operators and crews take the blame after a major failure, when in fact the most serious errors took place long before and were the fault of designers or managers whose system would need superhuman performance from mere mortals when things went wrong."

As numerous analyses indicate, the primary "design faults" that Chiles refers to were largely failures to properly coordinate the interactions between people and technology (Woods & Sarter, 2000) in the development as well as the deployment of systems. Simply put, systems cannot be designed in isolation from the people who will ultimately use them, and the challenge of *Enterprise System Engineering (ESE)* is to move beyond the technology of systems or systems-of-systems to address the issue of *Human-System Integration (HSI)*.

1.2 Human-System Integration

The interactions between airline pilots and cockpit automation exemplify the challenges of HSI in ESE. Here, as in many domains, the role of the operator (pilot) has changed from direct manual control of various systems to supervisory control of automated systems (e.g., the Flight Management System and the autopilot). Clearly, if the air crew is to be successful in their new role of programming and monitoring these automated systems then they must have an adequate understanding of how the systems perform, and yet even expert pilots report difficulty in understanding and predicting the behavior of Flight Management Systems. This can lead to a loss of situation awareness and associated complacency (Sarter & Woods, 1992). This is especially problematic when automated systems encounter problems that systems designers did not and perhaps could not anticipate – because then the control of the aircraft is left in the hands of a pilot who may be "coming in cold off the bench".

Some would argue that the basic problem in command and control of complex systems is that error-prone humans are simply inadequate to handle the rigors of decision-making along with the vast amounts of available information, and "with just a little more automation we can eliminate the 'human error problem' entirely" (Christoffersen & Woods, in press). But the fact is that an enterprise is a complex interaction of people (e.g. soldiers, commanders, etc.) and processes as well hardware and software systems – and people will always be central players in enterprises because of their creativity, expertise, and adaptability. This is particularly so as the line between systems acquisition and systems operation is blurred by human beings who are constantly adapting themselves and their systems to emergent challenges of business, warfare, etc. Thus, we think the need for HSI in ESE will grow, not shrink.

1.3 Cognitive Engineering

Faced with the challenge of HSI in ESE, the question is: How can Enterprise Systems Engineering make the best use of people and systems in large-scale distributed and dynamic enterprises? We believe that the answer is to augment the practice of Systems Engineering with the methods of *Cognitive Engineering*.

Cognitive Engineering draws on a variety of disciplines, including Human Factors Engineering, Human-Computer Interaction, Decision Science, Cognitive Psychology, Computer Science, and other related fields. It has roots in Task Analysis, which identifies the key tasks or functions that are performed in a work domain and then systematically breaks each task into a series of lower-level tasks. Armed with such a task breakdown analysis, it is then possible to make engineering decisions about how to allocate functions between people and systems.

Here it is important to distinguish between *Behavioral* Task Analysis and *Cognitive* Task Analysis, since Cognitive Engineering is most concerned with the latter. Behavioral Task Analysis is concerned with actions (behavior) that can be directly observed, such as moving a dial or flipping a switch, and it is most often used to measure quantities like time-to-completion or total throughput in a given time. Cognitive Task Analysis moves beyond observable behavior to measure and model the mental activities (cognition) that drive observable behaviors, and it can be used to assess quantities like throughput as well as quality. For example, Cognitive Task Analysis can be used to assess the potential for human errors in information processing, and thereby serve as a basis for designing decision support systems.

The goal of Cognitive Engineering is to develop systems, training, and other products that support cognitive functions in decision-making, situation assessment, course-of-action selection, resource allocation and other information processing tasks. Some design questions addressed by Cognitive Engineering include: What information should be provided to system operators? How should the display be formatted so it is congruent with operator goals and decision-making objectives? How can tasks be effectively distributed across team members and system automation? How can systems support humans so that human-system performance is better than either systems or humans could achieve in isolation?

In the remaining sections of this paper we discuss how the methods of Cognitive Engineering can be used to address key problems in Systems Engineering. Section 2 begins with a survey of Cognitive Engineering techniques in the form of a *Methods Matrix*, which maps the various methods of Cognitive Engineering (in rows) to potential uses in Systems Engineering (columns). Section 3 provides a more focused look at how Cognitive Engineering methods have been applied to Systems Engineering problems in the domain of air traffic control. Section 4 offers a broader look at the potential applications of Cognitive Engineering to Systems Engineering, via short summaries of all the rows and columns of the Methods Matrix.

2. Cognitive Engineering Methods and Systems Engineering Uses

To begin, we present a survey of various methods in Cognitive Engineering and their potential uses in Systems Engineering. The methods of Cognitive Engineering are placed into five categories, based on the focus and purpose of each method. The five categories, illustrated along with several subcategories in Figure 1, are as follows: Modeling Cognitive Processes, Modeling Behavioral Processes, Describing Cognitive and Behavioral Processes, Modeling Erroneous Actions, and Modeling Human-Machine Systems. Each individual method was assigned to a single category/subcategory, and although some methods might be placed in several categories we chose the one that we believed was the best fit.

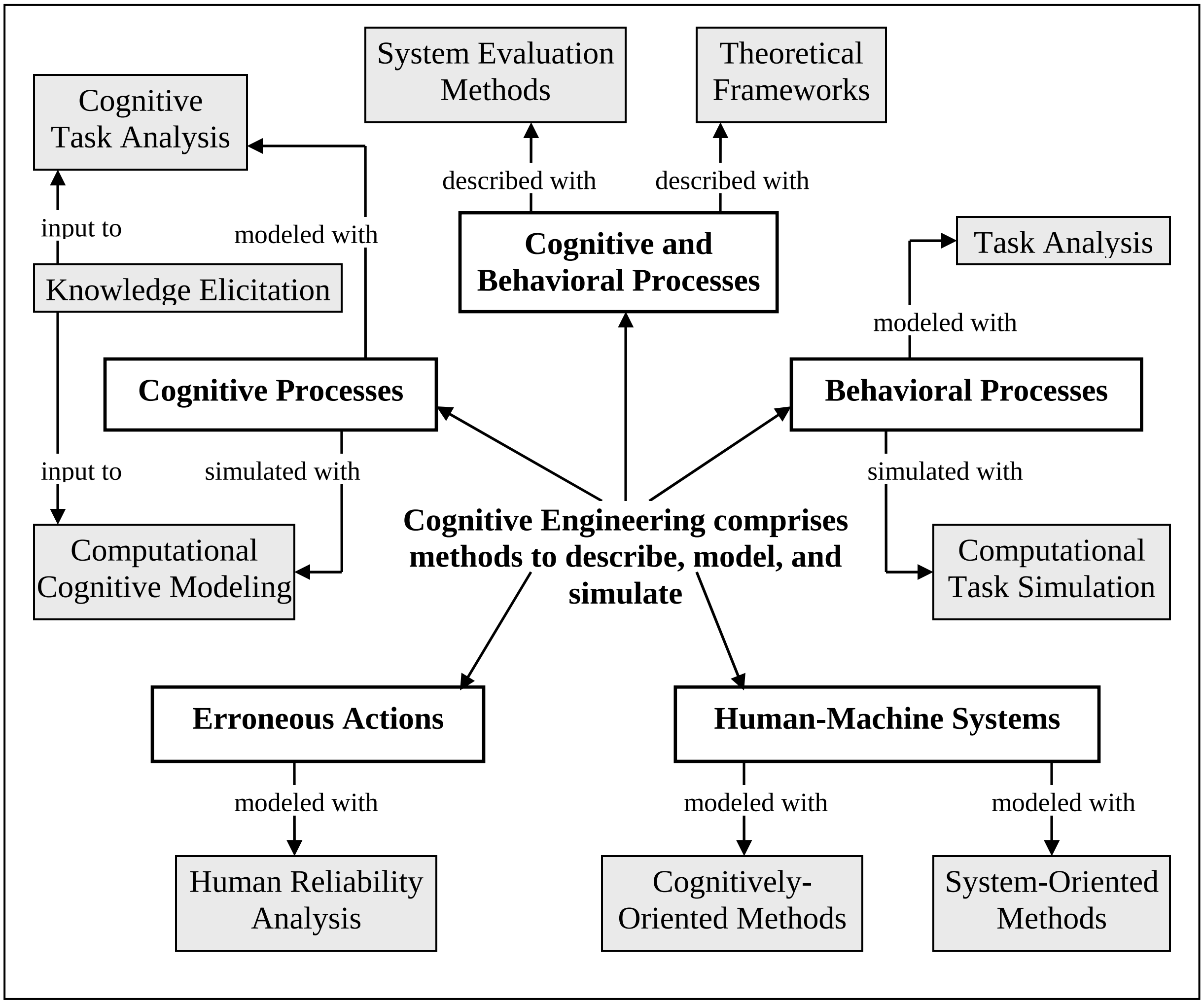

Figure 1: Classification of Cognitive Engineering methods.

The individual methods of Cognitive Engineering, organized by these categories/subcategories, form the rows of our Methods Matrix shown in Figure 2. The columns of the Methods Matrix represent phases of the Systems Engineering design life cycle, taken from Dugger, Parker, & Winters, 1999. Like the methods (rows), the phases (columns) might be listed differently and our scheme is intended to capture the major distinctions that we feel are important.

The cells of the Methods Matrix are shaded to indicate how useful the Cognitive Engineering method (row) is for the Systems Engineering problem (column). Black means “very useful”, gray means “somewhat useful” and white means “not too useful”. These judgments were based on a detailed review of the methods and how they have been used by others, where black or gray was assigned when a Cognitive Engineering method (row) was actually or could potentially (in our judgment) be usefully applied to the Systems Engineering problem (column). Further details on our judgments are available elsewhere (Bonaceto, 2003; Bonaceto & Burns, in press). Further description of the methods (rows) and phases (columns) are provided in Section 4 below.

Modeling Cognitive Processes

	Engineering Phase Method	Concept Definition	Requirements Analysis	Function Analysis	Function Allocation	Task Design	Interface & Team Development	Performance, Workload, Training Estimation	Requirements Review	Personnel Selection	Training Development	Performance Assurance	Problem Investigation
Cognitive Task Analysis	Applied Cognitive Task Analysis	■	■	▒	▒	■	■	▒	▒	■	■	▒	▒
	Critical Decision Method	■	■		▒	▒	▒	▒		▒	■	■	■
	PARI					▒	▒	▒		■	■		
	Skill-Based CTA Framework	▒	▒		▒	■	▒	▒		■	■	▒	
	Decompose, Network, & Asses					▒	▒	▒		▒	■		
	Task-Knowledge Structures					■	■	■			■	▒	
	Goal-Directed Task Analysis	▒	■	■	■	■	■	■	▒		■	▒	▒
	Cognitive Function Model			■	■	■	▒	■			■		
	Cognitively Oriented Task Analysis					▒	▒	▒		■	■		
	Hierarchical Task Analysis	▒	■	■	■	■	■	■	▒		■	▒	▒
	Interacting Cognitive Subsystems					■	■	■			■	▒	
	Knowledge Analysis and Documentation System					■	■				■		
Knowledge Elicitation	Unstructured Interviews	■	■			■	■		■		■	■	■
	Structured Interviews	■	■			■	■		■		■	■	■
	Step Listing	▒	▒			■	■	■			■		
	Group Interview	■	■			■	■		■		■	■	■
	Questionnaires	■	■			■	■		■		■	■	■
	Teachback	▒	▒			■	■	■			■		
	Field Observations/ Ethnographic Methods	■	■		■	■	■				■	■	■
	Twenty Questions	▒	▒	▒	▒	■	■	▒	■	▒	■		
	Discourse/Conversation/ Interaction Analysis	▒	▒			▒	■					■	
	Activity Sampling								▒			■	▒
	Think-Aloud Problem-Solving	▒	▒	▒	▒	■	■	▒	■	▒	■		
	Retrospective/Aided Recall	▒	▒	▒	▒	■	■	▒	■	▒			
	Interruption Analysis					■	■		■		■	■	
	Shadowing Another					■	■		■		■	■	
	Simulators/Mockups					■	■	■	■	■	■	■	
	Exploratory Sequential Data Analysis					■	▒	▒	■	▒	■		
	Minimal Scenario Technique	▒	▒	▒	▒	■	■	▒	■	▒	■		
	Critical Incident Technique	▒	▒		▒	▒	▒	▒				■	■

Modeling Cognitive Processes

	Engineering Phase Method	Concept Definition	Requirements Analysis	Function Analysis	Function Allocation	Task Design	Interface & Team Development	Performance, Workload, Training Estimation	Requirements Review	Personnel Selection	Training Development	Performance Assurance	Problem Investigation
	Cloze Technique	▒	▒	▒	▒	■	■	▒	■	▒	■		
	Critiquing	▒	▒	▒	▒	■	■	▒		■	■	▒	
	Crystal Ball/Stumbling Block	▒	▒	▒	▒	■	■	▒	■	▒	■		
	Table-Top Analysis	■	■	▒	▒	■	■	▒	■	▒	■		
	Wizard of Oz	■	■								■		
	Decision Analysis	▒	▒		▒	▒	▒				■		
	Rating and Sorting Tasks	■	▒			▒	▒			■	■		
	Magnitude Estimation	■	▒			▒	▒			■	■		
	Repertory Grid	■	▒			▒	▒			■	■		
	P Sort	■	▒			▒	▒			■	■		
	Q Sort	■	▒			▒	▒			■	■		
	Hierarchical Sort	■	▒			▒	▒			■	■		
	Cluster Analysis	■	▒			▒	▒			■	■		
	Multi-dimensional Scaling	■	▒			▒	▒			■	■		
	Likert Scale Elicitation	■	▒			▒	▒			■	■		
	Structural Analysis	■	■			▒	▒			■	■		
	Conceptual Graph Construction	■	▒			▒	▒			■	■		
	Concept Mapping	■	▒			▒	▒			■	■		
	Diagramming	■	▒			▒	▒			■	■		
	Laddering	■	▒			▒	▒			■	■		
	Influence Diagram Construction											■	■
Computational Cognitive Modeling	Keystroke Level Model					▒	▒	■	■			■	
	CMN-GOMS					■	■	■	■			■	
	NGOMSL					■	■	■	■			■	
	CPM-GOMS					■	■	■	■			■	
	Cognitive Analysis Tool					■	■	■	■			■	
	COGNET					■	■	■	■			■	
	COGENT					▒	▒	■	▒			■	
	ACT-R					■	■	■	▒		▒	■	
	Soar					■	■	■	▒		▒	■	
	EPIC					■	■	■	▒	■	■	■	
	Apex					■	■	■	▒		▒	■	
	MIDAS					■	■	■	▒		▒	■	
	SAMPLE					■	■	■	▒		▒	■	
	OMAR					■	■	■	▒		▒	■	

Modeling Behavioral Processes

	Engineering Phase / Method	Concept Definition	Requirements Analysis	Function Analysis	Function Allocation	Task Design	Interface & Team Development	Performance, Workload, Training Estimation	Requirements Review	Personnel Selection	Training Development	Performance Assurance	Problem Investigation
Task Analysis	Behavioral Task Analysis				▒	■	■				■	■	
	Operational Sequence Diagrams					■	■				■	■	
	Timeline Analysis					■	■					■	
	Operator Function Model			■	■	■	▒	■			■	■	
	Link Analysis					■	■					■	
Task Simulation	IMPRINT		▒	▒	▒	■	■	■	■			■	
	CART		▒	▒	▒	■	■	■	■			■	
	Micro Saint		▒	▒	▒	■	■	■	■			■	
	WinCrew		▒	▒	■	■	■	■				■	
	IPME		▒	▒	▒	■	■	■				■	

Describing Cognitive and Behavioral Processes

	Engineering Phase / Method	Concept Definition	Requirements Analysis	Function Analysis	Function Allocation	Task Design	Interface & Team Development	Performance, Workload, Training Estimation	Requirements Review	Personnel Selection	Training Development	Performance Assurance	Problem Investigation
System Evaluation Methods	Heuristic Evaluation								■			■	▒
	Walk-throughs/ Cognitive Walk-throughs/ Talk-throughs						■	■	■			■	■
	Formal Usability Studies						■	■	■		■	■	■
	Rapid Prototyping	■					■		■				
	Storyboarding	■					■		■				
	Interface Evaluation Surveys						■						
	Ergonomic Checklists			▒	■	■	■	▒	■				■
	Contextual Inquiry	■	■		■	■	■				■	■	■
Theoretical Frameworks	Activity Theory	■	■			■	■				■		
	Situated Cognition	■	■		■	■	■				■		
	Distributed Cognition	■	■		■	■	■				■		
	Naturalistic Decision Making	■	■	▒	▒	■	■	▒	▒	■	■	▒	■

Modeling Erroneous Actions

	Engineering Phase / Method	Concept Definition	Requirements Analysis	Function Analysis	Function Allocation	Task Design	Interface & Team Development	Performance, Workload, Training Estimation	Requirements Review	Personnel Selection	Training Development	Performance Assurance	Problem Investigation
Human Reliability Analysis	Event Tree Analysis											■	■
	Fault Tree Analysis											■	■
	Failure Modes and Effects Analysis											■	■
	Barrier Analysis					■	░					■	
	Hazard and Operability Analysis											■	
	Management Oversight Risk Tree											■	
	Work Safety Analysis										░	■	
	Confusion Matrices											■	
	Operator Action Event Tree											■	
	Generic Error Modeling System											░	■
	Cognitive Reliability and Error Analysis Method					■	░	░				■	■

Modeling Human-Machine Systems

	Engineering Phase / Method	Concept Definition	Requirements Analysis	Function Analysis	Function Allocation	Task Design	Interface & Team Development	Performance, Workload, Training Estimation	Requirements Review	Personnel Selection	Training Development	Performance Assurance	Problem Investigation
Human-Machine Systems Analysis	Cognitive Work Analysis	■	■	■	■	■	■	■	■	■	■	■	■
	Applied Cognitive Work Analysis	■	■	■	■	■	■	■	■	■	■	■	■
	Cognitive Function Analysis	░	■	■	■	■	■	■	■	░	■	■	
	COADE Framework	░	■	■	░	■	■	■	■	░	■	░	
	Perceptual Control Theory Approach	░	■	■	■	■	■	■	■	░	■	░	
	Information Flow Analysis			■	░							■	
	Functional Flow Analysis			■	░								
	Function Allocation			░	■								
	Mission and Scenario Analysis	■	■										
	Signal Flow Graph Analysis		■	■		■					■	■	

Figure 2. Matrix of Cognitive Engineering Methods and Systems Engineering phases (Online version: http://mentalmodels.mitre.org/cog_eng/ce_sys_eng_phases_matrix.htm).

3. Examples of Cognitive Engineering in Air Traffic Control

In this section we discuss specific applications of Cognitive Engineering methods to Systems Engineering problems in the domain of air traffic control, which is a complex enterprise comprising many agents with various goals and concerns. Here we focus on design efforts to increase capacity, i.e., to increase the number of aircraft able to take off and land in a given period of time. Such efforts typically include traffic control procedural revisions, or the construction or reconfiguration of taxiways and runways. Physical reconfigurations almost always require procedural revisions, and together these changes will affect the cognitive workload and decision performance of air traffic controllers – hopefully for the better, but then that is the engineering question. Therefore, before changes are made, it is important to predict the effects of system improvements on human performance, to ensure that human-system performance will in fact be improved. Below we discuss how The MITRE Corporation used three types of Cognitive Engineering of methods to address the Systems Engineering challenges associated with increasing capacity at a major airport (Bonaceto, Estes, Moertl, & Burns, 2005).

Referring to the columns of the Methods Matrix (Figure 2), the Systems Engineering challenges here involve Performance, Workload, and Training Estimation (How will the changes impact performance?); Task Design (How will tasks be performed in the future configuration?), and Requirements Review (Will controller performance be adequate to meet the new demands?). To address these challenges, we adopted a suite of Cognitive Engineering methods that includes Cognitive Task Analysis, Computational Cognitive Modeling and Critical Incident Analysis.

3.1 Cognitive Task Analysis.

To gain insight into how air traffic control tasks will be performed with a new set of procedures, we began with an initial Cognitive Task Analysis (CTA) to determine how tasks are currently performed. The CTA identified specific tasks for each controller position, based on review of airport procedural documentation as well as interviews with controllers. For each task, a decision inventory was constructed to enumerate all of the decisions that controllers must make to accomplish the task. This included a “control loop”, which models how a task is typically sequenced and when each decision making event occurs. The decision making events in a control loop were further expanded by identifying the artifact (tool or system, such as a radar display) used to support the decision and by characterizing the type of decision itself (e.g., a planning decision).

To validate and extend the CTA, field observations were then conducted in the existing air traffic control tower. These observations captured data about controller decision making events identified in the CTA under a variety of airport operating conditions (e.g., day, night, high and low traffic loads, adverse weather); for example when events occur, how events occur in sequence with other events, and what systems are used to facilitate them. In making these observations, we also noted how controllers adopted different decision making strategies in different contexts, and how shifts in strategies affect the time required to perform each task.

3.2 Computational Cognitive Modeling.

As a complement to the CTA, Computational Cognitive Models (CCM) were constructed using a Natural “Goals Operators Methods and Selection rules” Language (NGOMSL) method described by Estes & Masalonis (2003). These models used working memory load, which refers to the number of distinct pieces of information a controller must keep in memory during each step of a task execution sequence, as a measure of cognitive workload. The resulting cognitive models predicted cognitive workload, as well as the total amount of time required to complete each task. The field observations (discussed above) were used to validate these task time predictions.

These baseline models were then modified to predict performance under revised sets of procedures proposed for the airport's future configuration. This allowed comparisons of cognitive workload and task execution time using a variety of designs and procedures, and shed light on which modifications were likely to yield the best performance in the future airport. The Computational Cognitive Models were also used to answer questions about the number of controllers needed to perform each task in order for cognitive workload to be kept at a moderate level.

3.3 Critical Incident Analysis.

While Cognitive Task Analysis and Computational Cognitive Modeling provided many insights into controller performance, a remaining concern was the non-routine incidents (near-accidents) known as *runway incursions*. To address this concern we used Critical Decision Analysis of factors that led to runway incursions at the airport in its current configuration and, by extension, in the future configuration. This analysis considered the key factors that affect runway safety at both the current airport and at another airport configured similarly to the future airport design. Having identified a set of key factors, we then interviewed controllers at each airport, and supplemented the analysis by documenting controllers' insights relative to the key factors.

Taken together, the results of the Cognitive Task Analysis, Computational Cognitive Modeling, and Critical Decision Analysis were ultimately used to inform the design of a set of scenarios for high fidelity simulations in which air traffic controllers would perform their tasks. These simulations are very costly to build and run, and our Cognitive Engineering efforts helped to design focused scenarios that could illuminate whether or not the proposed modifications to systems and procedures would in fact improve human-system performance.

4. Detailed Discussion of the Methods Matrix

In this section we provide more details on the Cognitive Engineering methods (rows) and Systems Engineering phases (columns) of the Methods Matrix.

4.1 Cognitive Engineering Methods (Rows).

Modeling Cognitive Processes: The methods in this category are concerned primarily with modeling the knowledge and cognitive activities used to perform tasks in a work domain. Subcategories in this family include Cognitive Task Analysis, Computational Cognitive Modeling, and Knowledge Elicitation.

Cognitive Task Analysis seeks to represent the knowledge and cognitive activities operators utilize to perform complex tasks in the work domain (Schraagen, Chipman, & Shalin, 2000). This is most useful in developing training programs and performance measures, establishing criteria to select people for certain jobs, and providing insight into the types of support systems that people may need, as well as the algorithms such support systems may utilize.

The products of Cognitive Task Analysis are typically descriptive models, while the family of methods referred to as Computational Cognitive Modeling produces more detailed models of how humans perform complex cognitive tasks. Such models, which can run a computer, might provide insight into how well a proposed system will support operators by predicting operator performance and workload under a variety of situations, along with estimates of the time required to learn and perform a cognitive task.

Knowledge Elicitation methods, which provide input to both Cognitive Task Analyses and Computational Cognitive Modeling, are used to determine the knowledge required to perform work tasks. The think-aloud or process tracing technique is a common knowledge elicitation technique in which the operator thinks aloud while actually performing some task or solving a problem. The procedure generates a protocol (a recording of the operator's deliberations, possibly including actions the operator took, what the operator was looking at, etc.) that can be transcribed and analyzed to uncover information about the operator's reasoning sequences and goal structures.

<u>Modeling Behavioral Processes:</u> The methods in this category are concerned primarily with modeling and simulating sequences of behaviors, including rule-based decisions that affect when particular sequences are activated and how sequences interact. While these methods are not well suited for analyzing highly cognitive tasks, they can identify tasks that are cognitively demanding and that therefore require further analysis. Subcategories include <u>Task Analysis</u> and <u>Computational Task Simulation</u>.

Task Analysis includes methods for producing detailed descriptions of the way a task is currently performed or could be performed. A typical Task Analysis yields a temporally ordered sequence of actions necessary to achieve a task, along with duration estimates of each action. Further analysis can be used to predict the total time required to perform a task using the resources that have been allocated, thus ensuring adequate task performance or mandating a change in the system design so that certain tasks can be completed in the desired amount of time (Kirwan & Ainsworth, 1992).

Computational Task Simulation techniques are the analog of Computational Cognitive Modeling, but model only the observable actions necessary to perform tasks rather than the underlying cognitive activities that drive task performance. The simulations can dynamically "run" tasks in real or fast time as a way of estimating complete cycle times, error likelihoods, workload, and accuracy (Kirwan & Ainsworth, 1992).

<u>Describing Cognitive and Behavioral Processes:</u> These methods "describe" how people perform work tasks, so they are generally less formal than the "models" considered above. The methods examine how operators use the tools they currently have available to them to perform tasks in the work domain, typically when there is already a system (or prototype) in place that is to be evaluated and improved. Subcategories in this family are <u>System Evaluation Methods</u> and <u>Theoretical Frameworks</u>.

System Evaluation Methods evaluate how operators interact with existing or proposed systems. They aim to assess how easy a particular system is to learn and use, and how well the system supports the tasks that operators perform. These methods are typically used in an iterative fashion to test and refine a proposed system design, evolving it from a prototype to a final design. A commonly used method in this group is the usability study, where operators are observed or videotaped while they perform tasks using a proposed system in a controlled environment. By observing many operators performing the same tasks under such controlled conditions, it is possible to identify aspects of the human-system interface that require improvement.

Theoretical Frameworks are perspectives about how people perform cognitive work. They can help focus knowledge elicitation efforts by positing the important aspects of worker-technology interaction that one should take into account. A well-known of the Theoretical Framework is Naturalistic Decision Making (NDM), which focuses on how people make judgments and decisions in environments that have high-stakes, multiple players, ill-defined goals, are uncertain and dynamic, and time pressured. A more specific theory is Klein's Recognition Primed Decision Model (RPD), which asserts that human decision makers select courses of action primarily based on recollection of past experiences, i.e., "recognition" of how the current situation is similar to past experiences (Zsambok & Klein, 1996).

Modeling Erroneous Actions with Human Reliability Analysis: These methods are used for analyzing situations in which errors have happened, or might happen. The goal is to determine whether human errors will have serious consequences, and to quantify the likelihoods of various types of errors. A common method from this family is Fault Tree Analysis, which shows the various failures that would have to occur in order to cause an undesired event (e.g., an accident). A fault tree is constructed as a series of logic gates descending through subsidiary events to basic events, which may be human errors, hardware/software failures, or environmental events. With a fault tree, it is possible to determine likely sources of errors and construct barriers to prevent them (Henley & Kumamoto, 1981).

Modeling Human-Machine Systems with analysis at the whole-system level: These methods have the broadest focus on how the entire system, consisting of technology and people, works as a whole in order to accomplish the overall goals of the system. This category includes Cognitively-Oriented Methods, which focus on the cognitive demands that are imposed on people operating in work domains, and System-Oriented Methods, which focus on information flows between and among systems and humans.

A Cognitively-Oriented Method is Cognitive Work Analysis, which is similar to Cognitive Task Analysis but comprises five specific stages: Work Domain Analysis, Control Task Analysis, Strategies Analysis, Social Organization and Cooperation Analysis, and Worker Competencies Analysis. The work domain is modeled as a Function Abstraction Hierarchy, which shows goal-means relationships on different levels of the hierarchy, including functional purpose, abstract function, generalized function, physical function, and physical form. This is useful for addressing how the goals and constraints of the work domain shape the decisions and actions that are necessary to perform the work. The remaining phases of Cognitive Work Analysis are used to identify the tasks that must be performed to meet the goals of the domain, the cognitive strategies used to perform such tasks, how such tasks may be allocated among people and technology, and the cognitive skills operators need to perform the tasks (Vicente, 1999).

A System-Oriented Method is Functional Flow Analysis in which an analyst decomposes a system into the functions it must support. Function-flow diagrams are constructed to show the sequential or information-flow relationships between system functions, e.g., using Petri Nets as a modeling formalism to implement function-flow diagrams (Meister, 1989).

4.2 Systems Engineering Phases (Columns)

Concept Definition: At the outset of system design, the objective is to identify the system's mission and required capabilities, i.e., the reason for the system to exist. Cognitive tasks that are particularly challenging and that may require support systems may also be identified at this stage.

Requirements Analysis: After the system concept is defined, more detailed system requirements and specifications are then developed. Here Cognitive Engineering is concerned with human performance requirements, including usability and learnability requirements, and with human information needs and decision points.

Function Analysis: Next, the system functions needed to meet the mission requirements are defined. The function analysis includes all aspects relevant to inclusion of people in the system, especially the human functions that are needed to allow the system to function.

Function Allocation: At this stage the concern is with effectively distributing functions between humans and systems. This is based on performance and workload studies to determine optimal allocations, possibly through the use of simulation.

Task Design: The goal of this stage is to analyze how people would (and should) carry out the functions that have been assigned to them. Here Cognitive Engineering identifies task interactions and sequences as well as the possible strategies that people may employ.

Interface and Team Development: Once the roles and tasks of people (with respect to the system) have been determined, general concepts and specific designs for interfaces between these people, their system(s), and other people/systems are developed.

Performance, Workload, and Training Estimation: Given a proposed system design, the physical and cognitive workloads of individuals and teams are assessed. Small-scale or full-scale simulations may be particularly useful at this stage.

Requirements Review: Throughout the development process, the system design is reviewed with respect to its requirements (i.e., operational needs). Here the role of Cognitive Engineering is to evaluate the system with respect to its impact on human performance, including usability, learnability, and decision-making.

Personnel Selection: The goal of this phase is to establish the required human competencies to perform the work of the system.

Training Development: The goal of this phase is to develop effective training procedures that impart and assess knowledge and skills.

Performance Assurance: Once the system has been deployed, the goal of this phase is to ensure that it continues to function as intended. Capabilities and deficiencies of the operational system are examined and may lead to new system requirements. Here Cognitive Engineering focuses on how well the system and people work together.

Problem Investigation: In this phase the term “problem” refers to accidents or other incidents that occur after the system is deployed. The focus is on modifying the system itself and/or human training or procedures to prevent problem recurrence, often based on “root cause” investigations.

5. Conclusion

In this paper we discussed why Human-Systems Integration is of paramount importance to Enterprise Systems Engineering, and we proposed that methods of Cognitive Engineering could and should play a central role in the practice of Systems Engineering. We provided a broad survey of the Cognitive Engineering methods in the form of a Method Matrix, and summarized some specific applications to Systems Engineering problems in the field of Air Traffic Control. We believe that similar applications are both possible and necessary in other domains, and we suggest that doing so will advance the practice of Enterprise Systems Engineering. As such our Methods Matrix provides a roadmap for using Cognitive Engineering to improve Systems Engineering.

References

Bonaceto, C. A. (2003). *A survey of cognitive engineering methods and systems engineering uses* (MITRE Product No. MP 03B0000061). Bedford, MA: The MITRE Corporation.

Bonaceto, C., & Burns, K. (in press). A survey of the methods and uses of cognitive engineering. In R. Hoffman (Ed.), *Expertise out of Context*.

Bonaceto, C., Estes, S., Moertl, P., Burns, K. (2005). *Naturalistic decision making in the air traffic control tower: Combining approaches to support changes in procedures*. Paper presented at the 7th International Conference on Naturalistic Decision Making, Amsterdam, The Netherlands.

Chiles, J. R. (2001). *Inviting Disaster: Lessons from the Edge of Technology*. New York: Collins.

Christoffersen, K. & Woods, D. D. (in press). How to make automated systems team players. In E. Salas (Ed.), *Advances in Human Performance and Cognitive Engineering Research Volume 2*. New York: JAI Press/Elsevier.

Defense Science Board. (2005, January). *Defense Science Board Task Force on Patriot system performance, report summary*. (DTIC No. ADA435837).

Dugger, M., Parker, C., & Winters, J. (1999). *Interactions between systems engineering and human engineering*. Retrieved November 7, 2005 from the Office of Naval Research, SC-21 Science and Technology Manning Affordability Initiative Web site: http://www.manningaffordability.com/s&tweb/PUBS/SE_HE/SE_HE_Inter.htm.

Estes, S. L., & Masalonis, A. J. (2003). I see what you're thinking: Using cognitive models to refine traffic flow management decision support prototypes. *Proceedings of the 47th Annual Meeting of Human Factors and Ergonomics Society,* Denver, CO.

Henley, J., & Kumamoto, H. (1981). *Reliability engineering and risk assessment*. New York: Prentice-Hall.

Kirwan, B., & Ainsworth, L. K. (1992). *A guide to task analysis*. New York: Taylor and Francis.

Meister, D. (1989). *Conceptual aspects of human factors*. Baltimore, MD: The Johns Hopkins University Press.

Sarter, N. B. & Woods, D. D. (1992). Pilot interaction with cockpit automation: Operational experiences with the Flight Management System (FMS). *International Journal of Aviation Psychology, 2(4)*, 303-321.

Schraagen, J. M., Chipman, S. F., & Shalin, V. L. (Eds.). (2000). *Cognitive task analysis*. Mahwah, NJ: Lawrence Erlbaum Associates.

Vicente, K. J. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Mahwah, NJ: Lawrence Erlbaum Associates.

Woods, D. D. & Sarter, N. B. (2000). Learning from automation surprises and going sour accidents. In N. Sarter and R. Amalberti (Eds.), *Cognitive Engineering in the Aviation Domain*. Hillsdale, NJ: Erlbaum.

Zsambok, C. E., & Klein, G. (Eds.). (1996). *Naturalistic decision making*. New Jersey: Lawrence Erlbaum Associates.

Biographies

Craig Bonaceto is a Senior Information Systems Engineer in MITRE’s Center for Air Force Command and Control Systems, where he applies Cognitive Engineering methods to improve the design of Enterprise Systems in air traffic control and military command and control. He also performs research on human “mental models” to improve Human-System Integration. He holds a BS degree in Computer Science and Psychology from Rensselaer Polytechnic Institute.

Kevin Burns is a Principal Scientist in MITRE's Center for Air Force Command and Control Systems, where he performs research on "mental models" in order to improve the design of decision support systems in command and control applications. He holds MS and BS degrees in Nuclear Engineering from MIT, and also studied as a special student in MIT's departments of Cognitive Science and Media Arts and Sciences.

.