Moving Autonomy Forward Conference 2006
De Vere Hotel Belton Woods
Lincoln, UK

CAN CWA INFORM THE DESIGN OF NETWORKED INTELLIGENT SYSTEMS?

M.L. Cummings
Humans and Automation Laboratory, Massachusetts Institute of Technology

Can CWA Inform the Design of Networked Intelligent Systems?

M.L. Cummings

ABSTRACT

Cognitive work analysis, originally formulated for the process control domain, has seen increasing use in command and control (C2) domains. Process control applications of CWA occur in causal domains, in which the actual process and related limitations of natural laws constrain the system. However, C2 domains are intentional domains which are human-activity based, as opposed to process-control based. The goals of intentional systems are organizationally driven and not constrained by the laws of nature, which means there is an exponential increase in degrees of freedom for possible outcomes. This is particularly true for intentional domains that are characterized by time-pressured, dynamic problems with a high interdependency of human decisions which is characteristic of C2. While CWA has been applied to command and control systems with varying degrees of success, its applicability to intentional C2 domains remains unclear due to inherent limitations. Moreover, because CWA cannot represent embedded systems, its future use in C2 systems with an emphasis on developing intelligent technologies to support network centric C2 operations remains questionable. This paper will discuss the theoretical underpinnings of CWA and demonstrate that in its current state, CWA has significant shortcomings that make it ineffectual for the design of networked intelligent C2 systems.

INTRODUCTION

Originating from research in process control human-machine interaction, Cognitive Work Analysis (CWA) was developed to provide cognitive system engineers with a framework for analysis that identifies a system's goals and constraints, but also the impact of the constraints of the environment (Vicente, 1999). The end result of this analysis should be a set of information requirements that informs the design of some kind of human-computer interface (HCI). As depicted in Figure 1, the CWA methodology incorporates five phases of analysis: Work domain analysis, control task analysis, analysis of effective strategies, analysis of social and organizational factors, and identification of worker competencies. The concentration of the initial phases of CWA is primarily on system ecological elements with a purported gradual shift to more cognitive issues. The end result should be a product in which not only worker's interactions are understood, but also one in which process relationships relevant to supervisory control and decision making are revealed.

The current CWA methodology has been traditionally applied to process control decision support systems in causal domains such as power generation plants, in which the actual process and related limitations of natural laws constrain the system (Bisantz & Vicente, 1994; Hajdukiewicz & Vicente, 2002; Vicente, 1999; Vicente & Burns, 1995). In addition to causal domains, the CWA approach has been applied to intentional domains including specific military command and control systems (Burns, Bryant, & Chalmers, 2005; Cummings & Guerlain, 2003; Naikar, 2006; Potter, Gualtieri, & Elm, 2002), as well as manufacturing scheduling (Higgins, 1998). Intentional domains are those that are dominated by human intent, goals, perceptions, and decisions, Causal-based systems such as those in process control are dominated by their characterizing physical laws (of both the underlying processes and the automatic control schemes regulating them). Although all systems are ultimately governed by the laws of physics, intentional systems can be best understood in terms of human intent and goals, the organizational structures used to achieve them, and cost/benefit structures associated with "good" or often "good enough" performance.

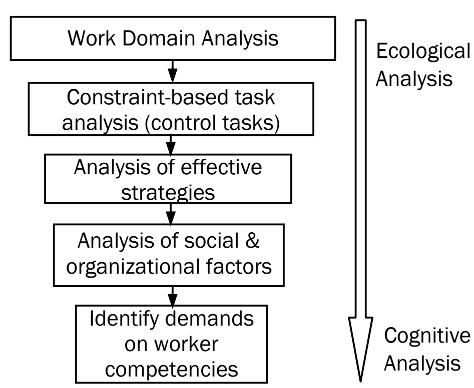

Figure 1: Five Phases of CWA

Military command and control (C2) systems are examples of intentional domains, where the physics of the problem (terrain, weather, firepower, etc) serve primarily as variables in the solution space, which is ultimately determined by human intent and goals. Such domains are characterized by time-pressured demands for reliable solutions under significant uncertainty with high-cost outcomes, against a similarly incentivized adversary whose intent is not easily modeled. Thus command and control problems cannot be viewed as simply another "process control" problem, and as a corollary, whether or not CWA techniques can truly be effectively applied to such intentional domains remains an open question.

CWA TOOLS

At the heart of the CWA methodology for systems analysis is the abstraction-decomposition space, which is the primary tool for the first stage work domain analysis but appears in the fourth stage as well (Vicente, 1999). The abstraction-decomposition space is used to represent the work domain, to include the goals and purposes of a system (Rasmussen, 1986). A two-dimensional modeling tool, the abstraction-decomposition space, shown in Figure 2, represents a whole/part breakdown of the system from the aggregate to the individual component (horizontal axis), and it also represents the means-ends relationships that describe the goals and priorities of the systems (vertical axis). For example, the objects are the means which allow the processes to occur, and the processes are the means that allow the general functions to occur and so on. The synthesis of the whole/part decomposition and the means/ends relationships cascade from left to right and top to bottom

In the abstraction-decomposition space, each cell can be expanded to detail further in-depth understanding of the work domain. This expansion, called an abstraction hierarchy, breaks each cell into the same means/ends five levels and further deconstructs the work domain. Whereas the abstraction-decomposition space demonstrates the "what" for a system, the abstraction hierarchy theoretically provides the additional "why" and "how." The notion of "when" in terms of what processes must occur at what time intervals is not addressed in CWA, which is a significant limitation for application in time-sensitive command and control domains.

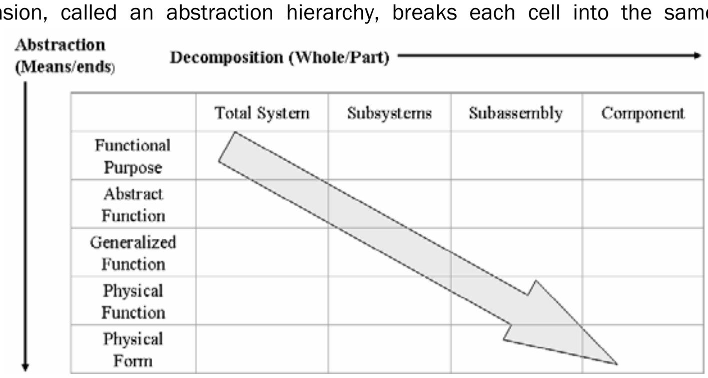

Figure 2: The Abstraction-Decomposition

The second phase of CWA, the control task analysis, provides a way in which to map decision processes through a decision ladder (originally conceived by Rasmussen (1986)). But this stage of CWA does not specify "who" makes these decisions, which is a critical deficiency when dealing with systems incorporating intelligent autonomy where agents (humans or automation) must have clearly specified roles. One of the overarching principles in the use of the decision ladder in CWA is that it specifically *not* address who should accomplish a control task, but specify instead general information requirements and knowledge states that somehow can be allocated to either humans or automation (Vicente, 1999).

For the last three stages of CWA, analysts are not given any specific direction or tools, even for traditional process control analyses, let alone non-traditional C2 problems. As will be discussed in a subsequent section, this is another problem with CWA. As of now, the abstraction decomposition and decision ladders are the only formalized analytical tools specified for use.

C2 APPLICATIONS OF CWA

In the published literature, there is only one example of a complete CWA (all five phases) for a C2 application, and this was shown to be significantly lacking for the design of a decision support system (Cummings, 2003). In this research effort attempting to design a decision support tool for single operator control of multiple Tactical Tomahawk missiles, the CWA process had to be significantly augmented with more traditional cognitive task analysis methods, as well as simulation models, to develop a functional interface. The remaining publicly available studies for the application of CWA in command and control domains only focus on very limited aspects (one or two phases) of CWA. For example, for four major projects sponsored by the Australian Department of Defence, only the first two phases of CWA were completed (at most) in each case (Naikar, 2006). In another C2 application analyzing the work domain of Canadian frigates, only the work domain analysis was conducted (Burns, Bryant, & Chalmers, 2005).

As previously noted, the general purpose of a CWA is to develop information requirements to develop HCIs, typically in supervisory control domains where operators only intermittently interact with an automated system to ensure safe and efficient operation. As such, one would expect that many HCIs would have been developed with the increasing use of CWA. However, the published literature is quite sparse in this regard. As mentioned previously, while a Tactical Tomahawk interface was developed partially using CWA (Cummings & Guerlain, 2003), very few command and control interfaces have been developed (or at least developed and reported on in the open literature). CWA was reportedly used to develop visualizations for air operation center (AOC) displays for the Air Force, but implementation or measurable performance improvements have not yet been reported (Gardner, 2004).

Instead of applying the CWA procedure for design, many researchers have chosen to develop interfaces based primarily on the abstraction-decomposition/hierarchy work domain analysis (otherwise formally known as ecological interface design (see Burns & Hajdukiewicz, 2004). One such example is the "Choose Combat Power" Display which purportedly supports military commanders in the stated overall functions of "complying with military law, minimizing collateral damage, complying with local laws and cultures, and a need to attain positive public perception" (Potter, Elm, Roth, Gualtieri, & Easter, 2002). It is important to note that the creation of this display was based on a modified CWA which only involved the first of the five original phases of CWA.

Despite the lack of tangible interfaces by which to evaluate the CWA methodology, there are numerous studies that purport to establish information requirements for the development of such a command and control interface. However, why an interface is not developed from these requirements is generally not discussed in these studies. Given the lack of actual interfaces, the CWA methodology could be judged by the quality and comprehensiveness of the requirements it generates. In one example, a CWA conducted to develop training system (i.e., simulator) requirements for an Australian F/A-18 provided the following requirements (Naikar & Sanderson, 1999):

- "Physical functionality: the F/A-18 training system must be capable of simulating the supersonic cruise capability of the F/A-18 aircraft and different levels of hostility and weather conditions."
- "Physical attributes: the F/A-18 training system must be capable of recreating the functionally relevant properties of VHF/UHF/HF radio channels and air and surface threats."

In another Australian application of CWA to the development of requirements for an airborne early warning aircraft, the construction of an abstraction hierarchy is credited with revealing this seemingly hidden requirement:

> *"[The abstraction hierarchy] shows that if deficiencies in the design of the radio voice links compromise the ability to exchange information and communicate, then the ability to establish, update, and disseminate the tactical picture, and to exercise control over friendly assets will also be compromised." (Naikar, 2006)*

In a study that claims to have developed the most comprehensive information requirements for a naval application using a work domain analysis, the following are a few examples of "between domain" requirements (the requirements that should be met considering a frigate, a possible contact, and the environment both are in) (Burns, Bryant, & Chalmers, 2005):

- "The team must understand the capabilities of the environment"
- "The ops room team must understand the contact's intent, constraints, and capabilities"
- "[The team] may want to disguise or to limit the enemy's access to information from the ship."

Lower level information requirements are then provided such as the need to display frigate location, contact location, and sea/waterway depth which are obvious requirements already met with current technology. One reported "new" information requirement uncovered by the work domain analysis is the need for contact visible marking information. Officers of the deck already know this information is critical, and generally gather this through lookouts, put in place to obtain such information.

In summary, while all these requirements are clearly true and important, they exist at such a high level as to not be useful or practical in the design or implementation of any contemplated HCI or decision support system. Furthermore, these requirements (and probably ones of greater design utility) could likely have been uncovered using a cognitive task analysis (CTA), a hierarchical task analysis, or even the functional decompositions used in traditional systems engineering processes (Blanchard & Fabrycky, 1998). Finally, the CWA approach to function decomposition is not new to the field of systems engineering and, in fact, under different names, has been in use much longer than CWA has been in existence. The following quote is from the most established systems engineering text in existence.

> *"Functional analysis is the process of translating system requirements into detailed design criteria, along with the identification of specific resources requirements at the subsystem level and below. One starts with an abstraction of the needs of the customer and works down to identify the requirements for hardware, software, people, facilities, data, or combinations thereof (Blanchard & Fabrycky, 1998)."*

Because many different methodologies exist which take a similar "organization chart" perspective on functional requirements, past CWA research has not made a convincing case that it provides better, more clear, or even more human-centered requirements. Moreover, as is the case with most requirement generation activities, it is often the skill, experience, and subject-matter expertise of the analyst(s) that drive the success of capturing the most accurate requirements. This is particularly true of CWA in which individual interpretation can vary widely and previous research has shown that successful CWA application is highly dependent on the subject matter expertise of the analysts (Jamieson, 2003).

PROBLEMATIC COMPONENTS OF CWA

The output of a CWA is reported by many to be a model of a system, more specifically, "a model that describes requirements that must be satisfied so that a System (sic) could behave in a new, desired way (Vicente, 1999)." This definition is problematic because models do not "describe" requirements. Models generalize interrelations from observed and/or simulated data, ultimately to *infer* indeterminate endogenous variables as a function of measurable exogenous variables, or to project future states from current states. While there are many different ways to model (qualitative descriptions, semantic networks, differential equations, etc.), tractable models can only represent interrelations of a small set of variables, and thus the usefulness of a model is typically inversely proportional to the number of variables it subsumes (Sheridan, 2005). Since models are general and abstract representations, at best models can aid an engineer in *developing* requirements, but models do not *describe* system requirements, especially those that are detailed.

In addition, abstraction decompositions and hierarchies are not models. These are representations of system elements and architectures, but they fundamentally lack the ability to infer (predict) one or more exogenous variables, via any sort of computational or algorithmic approach commonly used in the modeling and simulation community.. Jens Rasmussen, the originator of these tools, classifies them as a "framework for analysis and representation aimed at eliminating degrees of freedom in the set of behavior-shaping constraints…[which allows] converging on action alternatives (Rasmussen, Pejtersen, & Goodstein, 1994)." He further refers to the abstraction decomposition, a means-ends/part-whole representation, as a map for understanding how, what, and why a system is used.

The use of the abstraction decomposition/hierarchy to represent and map systems has been used extensively with varying degrees of success, but arguably it can be helpful in aiding designers attempting to understand a complex system in terms of components and their interactions ("boxes and arrows"). However, a topological representation is not the same as a model, and both engineers and psychologists should be more careful in applying terminology that is not appropriate. Because abstraction decomposition is the backbone of CWA, it is not a modeling tool; rather it is a domain analysis and potential system architecture mapping tool. This discussion is not meant to trivialize the use of these tools as they can be helpful, but calling them models is incorrect and misleading.

In addition to the theoretical debate as to whether or not CWA and/or the abstraction hierarchy can model a system, there are other potential limitations of CWA, which include: 1) Application to Intentional Domains, 2) Embedded Control Systems, 3) Revolutionary Systems, and 4) Ill-defined Analysis Process.

CWA has been shown to be effective for analyzing the human role in causal systems such as process control (Vicente (1999) but Lind (2003, 2004) would disagree), but its usefulness in intentional domains has been questioned (Cummings, 2003; Wong, Sallis, & O'Hare, 1998). Especially critical to command and control domains, the cause-and-effect relationships due to unanticipated events cannot be traced via the structural invariants provided by CWA. In addition, as mentioned previously, CWA has serious difficulty in addressing the concept of time in systems operations. In the first phase of abstraction decomposition, the analyst spends a great deal of time mapping out the what, how, and why relationships. While this may be effective for causal systems whose operations do not change dramatically over time, this is a limitation of CWA that makes it application to command and control (intentional) systems extremely limited, as is the inability of CWA to incorporate cause-and-effect tracings for unanticipated events in intentional domains. Any functional or design requirements that come from a CWA analysis for a command and control system should be evaluated carefully since there is no principled way within the analysis to address time-critical constraints.

A significant drawback to CWA is that the abstraction decomposition merges both process and control hierarchies, which are distinctly different and thus embedded control systems cannot be represented. This causes the subsequent representation to be both artificial and most likely incorrect. While current criticism of this particular CWA flaw has been directed towards its application to process control (Lind, 2003, 2004), this is a criticism that is even more applicable in command and control domains. There are countless embedded control loops found at all levels of C2 such as autopilot in planes, GPS navigation, electronic intelligence, radar-tracking solutions for fire-support, etc. Military platforms and C2 networks cannot exist without embedded control loops and with network-centric warfare on the immediate horizon, the presence of embedded control loops will only increase. Furthermore, another related embedded control problem, particularly in light of the application of intelligent autonomy in C2, is role allocation of control. One CWA assertion is that its decompositions represent system structure independent of any human, automation, event, or task goal. This inability to specify who should execute a task essentially excludes function and role allocation from the analysis, which is simply not a viable avenue in the development of an intelligent networked system including humans.,

Another problem with CWA is whether or not it is useful in the development of revolutionary systems. While purported to be a way to design revolutionary decision support systems (Vicente, 1999), CWA in its entirety can only be applied to existing systems (Cummings & Guerlain, 2003). CWA assumes an existing organizational structure, existing infrastructure, existing users, and clearly defined boundaries. However, in one other published case study, the first two phases only of CWA were used to design *teams* for a new system, but not develop requirements for technologies or team decision support systems (Naikar, Pearce, Drumm, & Sanderson, 2003). For decision support systems in revolutionary systems, CWA cannot be effectively used without other cognitive task analysis methodologies such as cognitive walkthroughs and simulations.

Lastly, CWA is a laborious process and the last 3 of its 5 phases are very vaguely defined. As previously described, many researchers and analysts will only complete the first two phases and label these results a CWA. Specifically the analysis of strategies, social, organizational, and cooperation analysis, and worker competencies phases do not have similar principled tools that are used in the domain and control task analysis phases. Despite numerous tools that could be applied to these area (social network theory, social judgment theory, decision trees, macrocognition, human performance modeling, etc.), it is not clear how and if these phases can provide any real contribution to the analysis of a system and the generation of any meaningful requirements for decision support. Moreover, it is in these phases where the emphasis of CWA is supposed to shift from ecological to cognitive, so in effect, the most important part of a *cognitive* work analysis is not accomplished and thus the cognitive needs of operators and teams are not supported. Significantly more research is needed in better defining the last three phases of CWA and demonstrating how they add overall value to the requirements process.

CONCLUSION

While CWA has merit in some areas, five major problem areas in the application of the CWA methodology to C2 systems have been identified and described here: 1) CWA is inadequate for the task of developing a true dynamic system model; 2) CWA has significant limitations when applied in intentional domains; 3) CWA does not account for embedded control systems; 4) CWA is best for describing existing systems, not designing revolutionary and novel systems; and 5) CWA's latter phases of analysis are ill-defined and lacking tools The future use of CWA in C2 systems, especially in the application of developing intelligent technologies to support network centric operations, remains questionable as a result, particularly because of the increasing use of automation and embedded systems, and the inability of the CWA methodology to address role allocations, time constraints, and the impact on operator cognitive processes.

While CWA in its current proposed form of five distinct stages has significant drawbacks in its application to C2 problem, this is not to say that elements of CWA are not useful. CWA components have been used in the past to develop evaluation standards/requirements for proposals (Naikar, 2006) and since these selective elements of CWA appear to capture functional high-level requirements, they may be more of a management/evaluation tool for command and control domains as opposed to a design tool. In addition to the use of CWA components as tools for analysts to more deeply understand a domain, one potential CWA application that is not addressed explicitly in the literature is that a main strength of CWA may be to help designers elicit knowledge from subject matter experts. This is not a trivial benefit. It is often very difficult for cognitive engineers to grasp all the nuances of human interaction in complex systems so if the CWA methodology aids the practitioner in both identifying critical relationships and mapping them in relation to other systems, then this is a valuable tool. However, caution must be taken in that these high-level representations can cause researchers to miss critical relationships, and should be augmented with other analysis tools as necessary.

As demonstrated in the numerous studies described previously, the abstraction decomposition/hierarchy is a CWA tool that many analysts find useful in developing abstract representations or maps of a complex system. Decision ladders, a tool originally proposed by Rasmussen (1986) which essentially constitute the second step of CWA, also can be used to generate decision support requirements (e.g., (Nehme, Scott, Cummings, & Furusho, 2006)). However, it is a repeated pattern throughout the published literature that the most effective uses of CWA are the abstraction hierarchy and decision ladders, tools that come from the sociotechnical research of Jens Rasmussen. One fundamental question that must be asked then is whether or not CWA is in effect a useful methodology in its entirety as it was originally proposed? If the final three CWA phases cannot be effectively applied (as many researchers have shown either explicitly or implicitly), then is CWA as a systems engineering methodology theoretically sound? In any case, the limitations discussed here in applying CWA to networked intelligent systems merit caution in its application to future command and control systems.

ACKNOLWEDGEMENTS

This work was supported in part by General Dynamics and special thanks to Greg Zacharias who provided both the inspiration and invaluable insight for this article.

REFERENCES

Bisantz, A. M., & Vicente, K. J. (1994). Making the abstraction hierarchy concrete. *International Journal of Human-Computer Studies, 40*, 83-117.

Blanchard, B. S., & Fabrycky, W. J. (1998). *Systems Engineering and Analysis* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Burns, C. M., Bryant, D. J., & Chalmers, B. A. (2005). Boundary, Purpose, and Values in Work-Domain Models: Models of Naval Command and Control. *IEEE Systems, Man, and Cybernetics Part A: Systems and Humans, 35*(5), 603-616.

Burns, C. M., & Hajdukiewicz, J. R. (2004). *Ecological Interface Design*: CRC Press.

Cummings, M. L. (2003). *Designing Decision Support Systems for Revolutionary Command and Control Domains.* Unpublished Ph.D. Dissertation, University of Virginia, Charlottesville.

Cummings, M. L., & Guerlain, S. (2003). *The Tactical Tomahawk Conundrum: Designing Decision Support Systems for Revolutionary Domains.* Paper presented at the IEEE Systems, Man, and Cybernetics Society Conference, Washington DC.

Gardner, C. (2004). AOC Strategy Visualizations. Dayton, OH: Air Force Research Laboratory.

Hajdukiewicz, J. R., & Vicente, K. (2002). Designing for adaptation to novelty and change: functional information, emergent feature graphics, and higher-level control. *Human Factors, 44*(4), 592-611.

Higgins, P. G. (1998). *Extending Cognitive Work Analysis to Manufacturing Scheduling.* Paper presented at the OzCHI '98, Adelaide, Australia.

Jamieson, G. A. (2003). *Bridging the Gap Between Cognitive Work Analysis and Ecological Interface Design* Paper presented at the Human Factors and Ergonomics Society 47th Annual Meeting, Denver, CO.

Lind, M. (2003). Making sense of the abstraction hierarchy in the power plant domain. *Cognition, Technology, & Work, 5*, 67-81.

Lind, M. (2004). *Means and Ends of Control.* Paper presented at the IEEE Systems, Man, and Cybernetics, The Hague.

Naikar, N. (2006). Beyond interface design: Further applications of cognitive work analysis. *International Journal of Industrial Ergonomics, 36* 423-438.

Naikar, N., Pearce, B., Drumm, D., & Sanderson, P. M. (2003). Designing teams for first-of-a-kind, complex systems using the initial phases of cognitive work analysis: Case study. *Human Factors 45*, 202-217.

Naikar, N., & Sanderson, P. (1999). Work domain analysis for training-system definition and acquisition. *International Journal of Aviation Psychology, 9*, 271-290.

Nehme, C. E., Scott, S. D., Cummings, M. L., & Furusho, C. Y. (2006, October 16-20). *Generating Requirements for Futuristic Heterogeneous Unmanned Systems.* Paper presented at the 50th Annual Meeting of the Human Factors and Ergonomic Society, San Francisco, CA.

Potter, S. S., Elm, W. C., Roth, E. M., Gualtieri, J. W., & Easter, J. R. (2002). Using Intermediate Design Artifacts to Bridge the Gap Between Cognitive Analysis and Cognitive Engineering. In M. McNeese & M. A. Vidulich (Eds.), *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosa!: A report produced under the auspices of The Technical Cooperation Programme Technical Panel HUM TP-7 Human Factors in Aircraft Environments (HSIAC-SOAR-2002-01)* (pp. 137-168). Wright Patterson Air Force Base: Human Systems Information Analysis Center.

Potter, S. S., Gualtieri, J. W., & Elm, W. C. (2002). *Case Studies: Applied Cognitive Work Analysis in the Design of Innovative Decision Support* (No. 2002-E02). Pittsburgh: Cognitive Systems Engineering Center, ManTech Aegis Research Corporation.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction*. New York: North-Holland.

Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive Systems Engineering*. New York: J. Wiley.

Sheridan, T. (2005, June 16-18). *Allocation of Attention: Fragments of a Model.* Paper presented at the Applied Attention: From Theory to Practice, Champaign, IL.

Vicente, K. (1999). *Cognitive Work Analysis. Toward safe, productive, and healthy computer-based work.* Mahwah, NJ: Erlbaum.

Vicente, K., & Burns, C. (1995). *A field study of operator cognitive monitoring at Pickering Nuclear Generating Station - B (CEL 95-04).* Toronto, Ontario, Canada: Cognitive Engineering Laboratory, University of Toronto.

Wong, W. B. L., Sallis, P. J., & O'Hare, D. (1998). *The Ecological Approach to the Interface Design: Applying the Abstraction Hierarchy to Intentional Domains.* Paper presented at the 7th Australasian Conference on Computer-Human Interaction, Adelaide, Australia.