CAN THE COGNITIVE ENGINEERING APPROACH PREVENT "NORMAL ACCIDENTS"? HOW DESIGN MIGHT IMPROVE SOCIETAL RESILIENCY TO CRITICAL INCIDENTS

Norman Groner, Ph.D.

Abstract

This paper examines how societies respond to critical incidents—defined as sudden, negative, unplanned, traumatic and transformative events—by designing better ways to prevent and mitigate future occurrences. Charles Perrow (1999), in his landmark book *Normal Accidents*, hypothesizes that accidents in tightly coupled interactively complex technological systems are inevitable, and that occasionally some of these accidents will invariably cascade into critical incidents. Cognitive engineering has developed largely as a means to prevent and mitigate technological systems accidents described by Perrow. Cognitive engineering approaches are discussed as responses to Perrow's examples of systems problems. In particular, problems associated with automation and situation awareness in complex systems are examined. The idea that design can enhance societal resilience by preventing and mitigating critical incidents is extended to the design of the organizational and political environments in which technological systems are embedded.

Introduction

At its best, a society responds to negative critical incidents[1] by enhancing its resiliency to repetitions of similar incidents. Societies do sometimes succeed in improving their abilities to prevent, mitigate, respond, and recover from critical incidents; presumably leading to a less fatalistic citizenry whose feelings of security lead to greater satisfaction and prosperity. One of contemporary society's foremost challenges is enhancing its resiliency to technological accidents associated with the complex and dangerous technological systems on which it is increasingly dependent. Societal resiliency depends on our ability to find new approaches that will allow us to cope with the potential of what, in his landmark book, Charles Perrow (1999) describes as "normal accidents"—system failures in technological systems that are so interactively complex and tightly coupled that on rare occasions, multiple and unexpected interactions inevitably cascade into catastrophic incidents with large losses of lives and material assets. Perrow (1999) discusses numerous such failures in nuclear power plants, petrochemical processing plants, commercial aviation, and maritime transportation.

In this paper, I use some of the examples provided by Perrow (1999) to examine how the developing discipline of cognitive engineering might have prevented these events from occurring or cascading out of control. Cognitive engineering is an important paradigm shift away from purely physical representations of engineered systems towards cognitive representations that emphasize the goals and limitations of people who interact with the technological systems.

About the author

Norman Groner is an associate professor in the Department of Protection Management at the John Jay College of Criminal Justice, the City University of New York. He has worked in the human factors field for 25 years, much of it in the area of cognitive factors related to fire safety, emergency planning and security management. Dr. Groner has Master of Science and Doctoral degrees in general psychology from the University of Washington.

The discipline of cognitive engineering (CE) focuses, in part, on designing socio-technical systems that are better able to prevent and respond to unforeseeable failures. In Perrow's (1999) view, these failures are the inevitable results of inherently risky technological systems that are tightly coupled and complexly interactive.

Perrow's Argument about "Normal Accidents" in Technological Systems

The *interactive complexity* and *coupling* of modern technological systems is continually increasing. This, in turn, makes it impossible to always predict system behaviors and respond appropriately, resulting in inevitable systems accidents. *Coupling* refers to the causal links in the behaviors of system components. Tight coupling means that effects *reliably* follow causes. Modern technologies require tight coupling, because system behaviors under normal operational conditions must be predictable. While tight coupling is generally desirable, it creates problems when technological systems are also interactively complex because negative events can quickly cascade toward unpredicted accidents with little opportunity to intervene.

Perrow (1999) defines "complex interactions" as "…unfamiliar sequences, or unplanned and unexpected sequences [that are] either not visible or not immediately comprehensible" (p. 78). *Interactive complexity* can be contrasted to linear interactiveness. Linear interactions progress in predictable and understandable ways; if a system component fails, the downstream results of that failure can be accurately predicted. However, interactively complex systems have components that serve multiple functions, are causally linked to multiple other systems, or are located in close proximity to components that are functionally unrelated. In interactively complex systems, component failures can cause effects that are unintended and unpredictable. Because the systems are tightly coupled, there is little time to try to understand and react to the unanticipated behaviors of the system.

Disruptions caused by social and physical environmental factors greatly compound the risk that technological systems will propagate unforeseeable behaviors. Examples of such uncertainties include natural disasters, social upheavals, production pressures and inadequate regulatory controls and oversight.

Because tightly coupled and interactively complex systems behave in unpredictable ways, especially under abnormal conditions, Perrow's (1999) theory of normal accidents hypothesizes that catastrophic failures are inevitable, albeit rare. Perrow (1999) recognizes that the frequency of catastrophic failures differs considerably among technological systems. He largely attributes the differences to trial-and-error, that is, the number of opportunities that various industries have had to improve the design of systems in response to serious accidents. He notes that the relatively long history of chemical processing makes systems accidents less likely than in nuclear power plants. However, an alternative explanation for such variance is that some industries devote more attention to the proactive design of their systems, resulting in much better records of responding to systems failures, even when the failures have never been experienced. The much enhanced attention that some industries devote to anticipating and responding to uncertainty is described by researchers in their discussion of high-reliability organizations (Sutcliffe, Obstfeld, & Weick, 1999).

As an alternative to Perrow's (1999) trial-and-error hypothesis, I propose that the development of cognitive engineering (CE) has been an important factor in counteracting the problems inherent to interactively complex systems. As suggested by Perrow (1999), unanticipated interactions may be impossible to prevent. However, CE researchers and theorists

offer important suggestions that enable systems to respond *adaptively* to such incidents; thereby potentially preventing events from becoming "critical" (i.e., interventions stop systems accidents from cascading towards catastrophic failures). Innovations from CE have been used to improve the operations of chemical process and nuclear power plants, oil tankers, airplanes and military equipment.

Perrow's (1999) analysis discusses important dangerous attributes of risky technological systems that involve complex interactions. This paper uses examples from Perrow's (1999) analysis to illustrate how the cognitive engineering practitioners have examined all of the features he describes and have developed design approaches that mitigate the impact of these types of problems.

The Cognitive Engineering Response to Interactively Complex Technologies

Perrow (1999) notes that system accidents are often misattributed to operator errors: "…if…the operator is confronted by unexpected and mysterious interactions among failures, saying that he or she should have zigged instead of zagged is possible only after the fact" (p. 9). Cognitive systems engineers agree; they refer to "design-induced errors" that are more accurately attributed to the poorly designed interfaces with which operators interact with complex systems than to "operator errors." Cognitive engineering is a rapidly evolving discipline that is part of the larger field of human factors that concerns itself with the design of interactions between people and the artifacts they build to accomplish goals. (In the 1999 edition of his book *Normal Accidents*, Perrow did not discuss the potentially ameliorating impact of CE. This is hardly surprising given that the discipline was still in its early development.)

Cognitive engineering[2] has largely evolved around the need to cope with the problems identified by Perrow (1999). Among other considerations, Woods and Roth (1988) described CE as a response to the need to design complex systems with multiple cognitive agents, including both machines and people. It is likely that CE will contribute in important ways to society's relationship with the complex hazardous technologies that can cause critical incidents.

Perrow (1999) raises two important issues: automation and interactive complexity. Cognitive engineering practitioners have worked on a wide variety of related problems. However, because Perrow (1999) specifically calls attention to these problems, this paper focuses on them to the exclusion of other problems tackled by cognitive engineers. Interactive complexity is more central to his argument, but I will discuss automation first, because the CE response is more straightforward, and because it introduces issues associated with controlling complex systems that I discuss in greater depth later.

The Cognitive Engineering Response to Automation Problems

Perrow (1999) discusses how automation is required to cope with the tightly coupled interactive complexity of many advanced technological systems. He explains that it is impossible to assure reliable human performance given the enormous amounts of information and rapid reaction times required to make adjustments in such systems. Instead of relying on people, system designers cope with system complexities by automating cybernetic (i.e., self-correcting) subsystems, thereby radically reducing the number of controls that the operators need to worry about.

Unfortunately, automation also adds significantly to the interactive complexities of the same systems, thereby creating additional hazards. In particular, Perrow (1999) identifies potentially catastrophic problems that are created when people do not understand how and why automated systems are taking certain actions. (In cognitive engineering jargon, these problems are called "mode errors.")

Cognitive engineers agree with Perrow (1999) that using automation to engineer human error out of systems can interfere with operators' attempts to diagnose system faults when they do occur. Operators sometimes fail to understand the behaviors of automated systems behaviors when those behaviors are unexpected. However, cognitive systems engineers are well-aware of the problems with automation, and have developed strategies to counteract the difficulties.

Lee (2006) explains that "mode errors are perhaps the most pervasive of the automation-induced errors…These arise when operators fail to detect the mode or recognize the consequences of mode transitions in complex automation" (p. 1573). Perrow (1999) provides examples of mode errors. He discusses the grounding of the tanker Torrey Canyon and the resulting disastrous oil spill in 1967: "When the helmsman received the order to come hard left on the wheel, nothing happened. The captain had forgotten to take it off automatic pilot the last time he turned it himself. He then threw the switch to manual so it could be turned…but it was too late" (p. 184). Perrow (1999) also discusses a mode error that contributed to the Three Mile Island Disaster. A technician had closed a valve to an emergency secondary cooling system that was unlikely to ever be needed. As reported, "…a valve in each pipe had been accidentally left in a closed position after maintenance two days before. The pumps [automatically] came on and the operator verified that they did, but he did not know that they were pumping water into a closed pipe" (Perrow, 1999; p.19).

Design features that correct for simple mode errors are straightforward; the interface must clearly show the mode in which the automated system is operating. A prominent indicator should have shown whether the automatic pilot on the Torrey Canyon was turned on or off, or better, feedback (e.g., a prominent message or signal) should have been provided when the wheel was turned while the automatic pilot was engaged. The mode error associated with the Three Mile Island mishap should have been circumvented by the use of displays that are close to other controls associated with system and that call attention to anomalous modes, for example, through the use of audible and visual alarms that highlight the current operating mode.

Unfortunately, in complex interactive systems, simple mode indicators may not provide the information that an operator needs to diagnose the precise nature of a systems fault and take corrective action. During the accident at the Three Mile Island nuclear power plant, indicator lights were available to show that the secondary cooling system valves were closed. However, one of the lights was obscured by a repair tag. More significantly, the operators always expected the valves to be open, so they did not look for closed valves as a source for the unexpected behavior of the system until eight minutes had passed and the reactor was seriously damaged. The sole reliance on indicator lights is clearly insufficient in complex systems. To deal with such complexity, CE practitioners have developed a more sophisticated approach, which is discussed in the next section.

Interactive Complexity

Perrow (1999) argues that the interactive complexity of many types of inherently dangerous technologies leads to systems behaviors that cannot be anticipated. Moreover, when

such behaviors occur, they cannot be comprehended quickly enough for human operators to make effectively adaptive responses.

Perrow (1999) gives credit where it is due: he explains that many potential disasters have been averted because people are inherently motivated to make sense of ambiguous situations and have extraordinary abilities to innovate responses in the face of unanticipated problems. These abilities cannot be replicated by computers. Because operators cannot be safely removed from the design of technological systems, the design challenge is to enhance operators' abilities to understand and respond appropriately to unexpected system interactions. In the next section, I discuss an approach to improving human understanding of interactively complex systems; that is, designing system interfaces that help operators maintain good situation awareness.

Situation Awareness

Endsley (1988) defines situation awareness as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". The mode errors previously discussed represent a relatively simple situation awareness problem, but complex systems require more sophisticated design approaches that assist people in understanding unexpected system behaviors. Operators of interactively complex systems can respond adaptively to unforeseen circumstances to the degree that they have high levels of situation awareness. The challenge in CE is to design system interfaces that facilitate high levels of situation awareness.

Perrow discusses events later in the Three Mile Island accident that were especially critical to the eventual negative outcomes of the emergency. Operators used a high pressure injection (HPI) system to try to cool the reactor vessel when the secondary cooling system failed, but they were unable to understand why the system was not responding as expected. Perrow (1999) explains: "After HPI came on, the operators were looking primarily at two dials… One indicated that the pressure in the reactor was still falling, which was mysterious because the other indicated that pressure in the pressurizer was rising—indeed, it was dangerously high. But they should move together and always had… If pressure is up in the pressurizer, and it is connected to the core, it should be up in the core" (p. 25).

Operators at Three Mile Island simply could not understand why the complex nuclear reactor was behaving in unexpected ways. One of the problems is that good situation awareness requires operators to understand and anticipate the complex interaction between vessel pressure and the relative levels of steam and water in a reactor, and these levels must be maintained within certain limits to avoid uncovering fuel rods. If left uncorrected, uncovered fuel rods result in a catastrophic meltdown of the reactor core. Sensors are unable to provide accurate direct measures of water and steam levels, so operators must derive the information from data about temperature and pressure. Vicente (2006) explains that nuclear power plant operators traditionally relied on steam tables to calculate water and steam levels:

> The procedure for evaluating reactor safety using a steam table requires quite a few steps. Operators have to memorize or record numbers… They may have to walk around to different locations in the control room. They have to look up values in a numerical table where, at a glance, each row looks like every other row. They have to do some unaided mental arithmetic, an error-prone process. In

> short, the psychological demands associated with using steam tables are not trivial. (p. 130-131)

In an emergency, such an exercise is too time consuming and error-prone. Reactors are now equipped with a Beltracchi's display (Beltracchi, 1987) or some equivalent that circumvents the procedure by providing operators with accurate situation awareness at only a glance. All the needed calculations are automated and displayed using an immediately comprehensible graphical display that shows operators the precise conditions over time in the pressurized reactor vessel relative to margins of safety. This type of display, unavailable at the time of the Three Mile Island disaster, provides operators with good situation awareness about conditions in the reactor vessel.

Augmented reality typically involves the use of real visual time displays of the real environment with computer generated features that improve situation awareness. A common example is a televised display of football games where lines are added to show the current scrimmage line and "first down" line. Innovations in augmented reality are rapidly being developed for use in aviation, both in cockpits (Aragon & Hearst, 2005) and air traffic control (Mackey, Fayard, Frobert, & Medini, 1998), where they are expected to further improve safety.

Global Situation Awareness

As noted earlier, humans have an innate ability to understand ambigouos situations and to diagnose faults. However, it is also an inherent characteristic of people to focus so narrowly on particular problems that they lose sight of the big picture. Cognitive systems engineers discuss the importance of maintaining global situation awareness while working on more narrowly focused problems. Endsley, Bolte, and Jones (2003) explain: "A frequent problem for situation awareness occurs when attention is directed to a subset of information and other important elements are not attended to, either intentionally or unintentionally" (p. 86). Fixating on particular problems while losing sight of the big picture was significant during the Three Mile Island accident, and has been a frequent contributor to many critical incidents caused by technological systems accidents.

Displays can be specifically designed to help people maintain an overview of a situation while attending to a more specific problem. Endsley et al. (2003) continue: "Attentional narrowing can be discouraged through the use of displays with…global SA. Global SA—a high level overview of the situation across operator goals—should always be provided" (p. 86).

Perrow (1999) provides a good example of how redesigned displays have improved the situation awareness facilitated by air traffic control displays. "Though the screens introduced in the 1970's were more 'indirect' in one sense, since they were a representation of information from the radar or transponder, the screen gave continuous read-outs of position, altitude, and direction. Most important of all, they did not require communication with the aircraft to determine altitude (and in earlier versions, communication to get heading and speed)" (Perrow, 1999; p. 160).

Conclusions

In my view, the prevention of critical incidents in interactively complex technological systems results more from proactive design than from design changes based on trial-and-error.

The cognitive engineering approach has yielded progress in reducing the likelihood of catastrophic accidents from technological systems that are inherently risky and interactively complex. It is difficult to argue that the *absence* of a catastrophic accident resulted from a paradigm shift towards cognitive engineering, but it is also difficult to argue that the extraordinarily improved record of aviation safety is the simple result of trial-and-error. USA Today reports that "the overall safety record in recent years is staggering. From 2000 through 2005, there were 46 million airline flights on U.S.-based airline jet aircraft. Only two crashed and killed passengers…" (Levin, 2006). Perrow (1999) attributes the extraordinarily strong safety record in commercial aviation to a much greater level of operating history in these technologies. But here Perrow (1999) contradicts himself. He argues that interactively complex systems inevitably behave in unpredictable ways. Therefore, improvement only results from the investigations of accidents. It does not seem credible that the trial-and-error approach that Perrow (1999) claims has improved the safety records of technologies with more operating time could yield the extraordinarily low accident rate experienced in commercial aviation, especially given the increasingly interactive complexity of each generation of advancing technological systems. Instead, the exemplary aviation record results largely from improved designs that anticipate failures, even when they have never happened. The contributions of human factors practitioners have been essential to these design innovations, and the resulting decrease in critical incidents (McFadden and Towell, 1999).

The improvement of aviation safety can be compared to disastrous maritime accidents. Filor (1994) concludes that there is a general trend to fewer accidents, although there is an argument that "there has been an increase in 'maritime disasters'…" (p. 159). Filor's review of maritime accidents concludes that like other high hazard industries, maritime accidents are largely attributable to human error, but that human factors engineering has not been applied to the same degree, resulting in less improvement. Given the far greater incidence of maritime accidents and the failure to improve the record of disastrous incidents, it is difficult to conclude that the relative improvement in aviation safety is wholly attributable to trial-and-error as opposed to proactive improvements in design, especially as in the area of cognitive engineering.

While designs that help people understand and react to unforeseen circumstances addresses Perrow's (1999) principal argument about technological systems, he also raises important social issues that profoundly affect safety. In this paper, I have focused on CE as a design approach that can prevent critical incidents in technological systems, but improved design can also alleviate problems related to organizational and political dysfunction discussed by Perrow (1999). In his book, Perrow (1999) admits that the hypothetically intractable problem of accidents in interactively complex and tightly coupled technologies cannot be attributed solely to a lack of experience. Instead, he implies that problems in the social environment of technological systems are largely to blame. For example, Perrow devotes considerable attention to the role of production pressures as a cause of disastrous technological accidents.

Vicente (2006) comes to a similar conclusion from a CE standpoint, and discusses human-systems interaction at several *systems* levels: physical, psychological (one aspect of which is discussed in this paper), team, organizational and political. He provides examples where human factors can, and have, contributed to improve designs at all these levels. Societies seem doomed to repeat history, except where societies have increased their resilience as evidenced by improved design at all these levels.

Nuclear power provides an instructive example. Three Mile Island was a technological accident in a nuclear power plant where poorly designed operator interfaces interfered with

operators' attempts to cope with the system's great interactive complexity. Three Mile Island fundamentally altered the public's view (at least in the United States) of nuclear power plants, resulting in a moratorium in the construction of new nuclear power plants. As a result of the accident, the human factors design of nuclear power plants has changed substantially in a successful (to date) effort to prevent additional critical incidents. (Systems accidents continue to occur in nuclear power plants, but better cognitive engineering designs help operators prevent the effects from cascading towards disastrous outcomes.) If we experience escalating fuel costs and a lack of critical incidents in nuclear power plants, change in public attitudes is likely to encourage the construction of new nuclear power plants. Finally, perhaps a lack of critical incidents in technological systems fundamentally changes the willingness of people to embrace technological change, resulting in more productive lives along with a general sense of security and well-being.

Notes

[1] A Critical Incident is a relatively brief occurrence involving injury, loss, conflict, discovery or change of significant proportion, usually unscripted and unanticipated, with the potential to alter existing societal norms. Critical incidents are usually traumatic, threatening the bonds of trust that bind communities, but may be positive, initiating historic consequents (Ochberg, 2009).

[2] "Cognitive engineering" is not a universally accept term among human factors professionals. It is used here as a catchall for studies of human-systems interactions described in terms of mental processes.

References

Aragon, C.R., and Hearst, M.A. (2005). Improving aviation safety with information visualization: A flight simulation study. CHI '05 Proceedings of the SIGCHI conference on Human Factors in computing systems. Los Angeles: Addison-Wesley.

Beltracchi, L. (1987). A direct manipulation interface for water-based Rankine cycle heat engines. *IEEE Transactions on Systems, Man, and Cybernetics SMC-17*: 478-487.

Endsley, M. R., Bolte, B., & Jones, D. G. (2003). *Designing for situation awareness: An approach to user-centered design.* New York: Taylor & Francis.

Filor, K. (1994). Marine accidents: Present trends and a perspective of the human element. Hulls, Hazards and Hard Questions – Shipping in the Great Barrier Reef. *Workshop Series.* Great Barrier Reef Marine Park Authority, 59-166. Retrieved from http://www.gbrmpa.gov.au/.

Hollnagel, E., & Woods, D. D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*. New York: Taylor & Francis.

Perrow, C. (1999). *Normal Accidents: Living with High-Risk Technologies*. Princeton, NJ: Princeton University Press.

Lee, J.D. (2006). Human factors and ergonomics in automation design. In G. Salveny (Ed.), *Handbook of Human Factors and Ergonomics, 3rd Edit*ion. Hoboken, NJ: Wiley.

Levin, A. (2006, June 30). Airways in USA are the Safest Ever. *USA Today*. Retrieved from http://www.usatoday.com

Mackey, W.E., Fayard, A., Frobert, L. and Medini, L. (1998). Reinventing the familiar: Exploring an augmented reality design space for air traffic control. *CHI '98, Proceedings of the SIGCHI conference on Human Factors in computing systems*. Los Angeles: Addison-Wesley.

McFadden, K. L. & Towell, E. L. (1999). Aviation human factors: A framework for the new millennium. *Journal of Air Transport Management*. *5*(4), 177-184.

Ochberg, F. (2009) The critical incident concept. Retrieved February 20, 2009, from the website of the Academy for Critical Incident Analysis at John Jay College. Retrieved from http://jcia.aciajj.org/about/the-critical-incident-concept.

Woods, D. D. & Ross E.M. (1988). Cognitive systems engineering. In M. Helander (Ed.), *Handbook of human-computer interaction*. North-Holland, New York.

Sheridan, T.B., & Parasuranam, R. (2006). Human-automation interaction. *In Reviews of Human Factors and Ergonomics, Volume 1*. Santa Monica, CA: Human Factors and Ergonomics Society.

Vicente, K. (2006). *The human factor: Revolutionizing the way people live with technology*. New York: Routledge.

Weick, K.E., Sutcliffe, K.M., & Obstfeld, D. (1999). Organizing for high reliability: Processes of collective mindfulness. *Research In Organizational Behavior*, *21*, 81-124.