Bennett, K. B., & Flach, J. M. (1994). When automation fails... In M. Mouloua & R. Parasuraman (Eds.), Human performance in automated systems: Current research and trends (pp. 229-234). Hillsdale, NJ: Lawrence Earlbaum Associates.

When Automation fails...

Kevin B. Bennett & John M. Flach
Wright State University
Dayton, Ohio, 45435
USA.

ABSTRACT

Automation has sometimes been viewed as an opportunity to replace human decision-making with machine algorithms or step-by-step procedures (e.g., automation, expert systems, SOP's, EOP's). However, there is a growing consensus that this approach, as the sole prescription for design, is fundamentally limited: circumstances will arise that cannot be foreseen, and, therefore, cannot be planned for (i.e., unanticipated variability). This is a primary reason for including a human operator in the system -- to respond to contingencies that were not anticipated by the designers. A complimentary approach (representation aiding) assumes that a problem solving role is inevitable for the human decision-maker and attempts to support him/her in that role. Graphic representations are used to present information about the system so that an individual's capability to assess situations and to devise solutions is improved.

We will focus on an approach to the design of representation aids that has been called "ecological interface design" (EID) (Rasmussen & Vicente, 1989). The emerging theory of EID relies heavily on Gibson's theory of ecological psychology and direct perception, and Rasmussen's concept of the abstraction hierarchy (this serves as a framework for describing meaningful dimensions of the ecology of human machine systems). The goal of this approach is to make the sources of constraint within a work domain visible to the human operator. This can be accomplished by mapping the nested hierarchy of work domain constraints onto spatial or geometric invariants within graphic displays.

INTRODUCTION

> What is increasingly clear is that the classic design goal of designing "idiot-proof" systems... is profoundly wrong... This is not to say that such design goals as searching out clarity in the user interface or consistency in the underlying organization of system functionality, are not desirable. Meeting these goals does not seem to prevent interpretative and other troubles in the use of informational systems. One can never anticipate and "design away" the exigencies, misunderstandings, and problems that will arise in people's use of systems; instead, we need to recognize and develop resources for dealing with the fact that the unexpected always happens. Said differently, we must widen our focus from designing for the avoidance of trouble to design for the management of trouble ... (Brown, 1986, pp. 464-465).

This quote refers to errors in traditional human computer interaction applications (HCI, e.g., word-processing programs, spreadsheets). However, the need to design for the "management of trouble" applies to any complex system (e.g., power plants, air traffic control, command and control), not just HCI applications. In fact, the opportunities for errors and accidents (as well as their consequences) are greatly amplified in many of these systems. Thus, there is a direct analogy to the "classic design goal" described in the previous quotation. Accidents in complex systems are usually interpreted to be the result of "operator error" and to make complex systems "idiot-proof" the operator has traditionally been replaced with various forms of automation (calculational, expert systems, automatic controllers, etc.). This approach, as the sole prescription for design, has also proven to be "profoundly wrong." An alternative design approach that recognizes the human component as an essential part of the system and seeks to support, rather than replace, the operator in this role will be discussed.

AUTOMATION

There is no question that automation can improve overall system performance by offloading tasks that strain limited cognitive resources of the operators (e.g., working or short term memory, attention), or tasks that require time-constrained system responses (e.g., reactor scram). On the other hand, history has made it very clear that automation will not eliminate errors or accidents. Automation can be designed to perform activities that can be planned for. This includes both routine tasks and classes of errors or accidents that can be expected due to the nature of the system. Thus, under normal circumstances automation does make the operators job easier by reducing the demands that are placed on them. However, in complex systems circumstances will arise that cannot be foreseen, and, therefore, cannot be designed away or planned for. Woods and Roth (1988) have referred to this as "unanticipated variability." Perrow (1984, pp. 82-83) discusses the inevitability and causes of "normal accidents:"

> These problems exist in all industrial and transportation systems, but they are greatly magnified in systems with many complex interactions. This is because interactions, caused by proximity, common mode connections, or unfamiliar or unintended feedback loops, require more probes of system conditions, and many more alterations of the conditions. Much more is simply invisible to the controller. The events go on inside vessels, or inside airplane wings, or in the space craft's service module, or inside computers. Complex systems tend to have elaborate control centers not because they make life easier for the operators, saving steps or time, nor because there is necessarily more machinery to control, but because components must interact in more than linear, sequential ways, and therefore may interact in unexpected ways.

Although automation may eliminate certain classes of errors, automation also makes the system more complex, thereby increasing the possibility of accidents and errors. For example, the system can fail in ways that it couldn't fail before: an autopilot system is set in the wrong mode and the flight path is not monitored, resulting in a crash. In addition, errors may become more difficult to detect and repair because automation adds an additional layer of complexity between the operator and the domain. The operator must understand the data that the automatic controller is acting upon, as well as the intent that lies behind that action. In effect, automation reduces the opportunity for direct interaction with the domain.

To summarize, automation does not offer the final solution for errors and accidents. Automation can introduce new classes of errors and can contribute to difficulties in diagnosis and repair when it creates distance between the operator and the work domain. When unanticipated variability occurs the demands on the operator are greatly increased, including the requirement for intelligent action. Thus, the need for the flexibility and adaptability of the human is increased, not eliminated, when automation is incorporated into the system. The emphasis of this article will be on representation aiding (graphic displays as decision support) as opposed to automation (generally speaking, computational integration). We see these approaches as complementary. Automation has greatly increased our capacity for measuring, differentiating, and integrating data. However, this data is of limited value if it remains hidden within the machine. Graphical interfaces will be critical for turning this data into information that the human operators can use to respond adaptively to meet the functional requirements within a complex work space.

REPRESENTATION AIDING

The use of dynamic, graphical representations holds great promise for increasing the capability of the human to deal with unanticipated variability. This article will describe (briefly) a functional systems approach to interface design. The focus is not on graphical forms, events, trajectories, tasks, or procedures. Instead, the focus is on domain constraints. Domain constraints refer to the multiple and interrelated goals that must be achieved within a domain, as well as the various means (or resources) that are available to achieve those goals. These constraints shape, but do not determine, behavior.

In complex, dynamic domains there are a large number of potential actions and behaviors that are available at any point in time. Typically operators have a standard sequence of behaviors (in some cases over learned to the point of being automatic) that are used to close these “degrees of freedom” (rule- and skill-based performance). Problems arise when the standard sequence fails and individuals must act in novel/creative/intelligent ways to meet the functional goals of the system (knowledge-based performance). When this occurs the expert will require the same type of decision support that designers need when developing the system. The expert will be involved in problem solving behavior and must consider the system from alternative conceptual perspectives to derive a plan to deal with unanticipated variability. What are the relevant goals? How can these goals be achieved? What are alternative system resources that might be used? What are the constraints on their use?

Traditional approaches have focused on behavior and standardization of behavior -- finding the one best way. Designing interfaces for "the one best way" results in a brittle system that quickly deteriorates as unanticipated variability occurs. The alternative to designing for the "one best way" is to design for adaptability. Where adaptability refers to the capacity to respond intelligently to the environmental (functional) constraints. Our assumption is that this capacity depends on information. That is, it depends on the ability to perceive ("see") the constraints. This is a fundamental assumption of our approach -- that the human will respond intelligently to constraints that are visible. Conversely, we assume that errors or inappropriate responses result from failure to perceive relevant constraints.

In complex systems the ability to "see" the constraints is threatened by both the presence of "too much data, and too little information." The challenge of interface design then, is to organize the data so that the amount of information is maximized. This involves consideration of correspondence (what constraints need to be made visible so that the interface is comprehensive) and coherence (how can large quantities of information be integrated so that it can be comprehended by a human operator).

CORRESPONDENCE

Correspondence refers to the issue of content --- what information is necessary in order to meet the functional demands of the work domain? In other words, what are the meaningful distinctions with regard to the domain constraints. A representation is complete if all meaningful distinctions in the work domain have corresponding distinctions within the representation. This requires at least a one-to-one mapping. A one-to-many mapping would also be possible in which some distinctions within the work domain are redundantly represented in the display space. However, in practice, representations are generally incomplete. Hard decisions must be made about what to include and what to leave out of the representation. Thus, the issue of correspondence arises--- what information about the work domain must be made available to the cognitive agent (whether human or computer).

Addressing the issue of correspondence requires a deep understanding of the work domain --- a comprehensive task analysis. Rasmussen's Abstraction and Aggregation Hierarchies (1986) are complementary theoretical frameworks for developing this deep understanding of the constraints within a work domain. The Abstraction Hierarchy is a description of the work domain in terms of a nested hierarchy of functional constraints that includes goals, physical laws, regulations, organizational/structural constraints, equipment constraints, and temporal/spatial constraints. The Aggregation Hierarchy is a description of the work domain in terms of part-whole relations.

Rasmussen's approach differs from traditional approaches to task analysis in two important ways. First, traditional approaches focus on trajectories through the work space. That is, they focus on observed or required behaviors --- on the actions. Rasmussen's approach, however, focuses on the constraints or landscape of the work space which provides the opportunities for action --- not on the specific actions. Second, traditional approaches to task analysis generally contain an implicit assumption that there is an "atomic" level of description at which the system can be completely understood (e.g., the "therblig" for manual work or the activities in Berliner's taxonomy for information processing). However, a critical assumption behind the abstraction hierarchy is that there is no single level that is privileged. The work domain must be understood at multiple levels of abstraction. Each level has emergent properties that cannot be understood as a simple function of the "atoms" at a lower level. Thus, the abstraction hierarchy highlights the difference between data (as independent observations) and meaning (as relationships across observations). Within the abstraction hierarchy, *relations over elements* at one level of abstraction appear as *elements* at a higher level of abstraction.

In complex domains, meaning is often relational. For example, the state of liquid in a process depends on a relation between pressure and temperature; the rate of change of mass in a volume depends on the relation between inputs and outputs; the stability of a chemical process will depend on complex relations between the relative amounts of different agents, the temperature, and the pressure. Thus, the representation is not complete if only the elemental data (e.g., the pressure and temperature) are represented independently. The relations themselves are also "objects of interest" and must be explicitly included in the representation if it is to be comprehensive.

Independent of whether you are designing graphical interfaces or computational expert systems, the issue of meaning and importance is critical. The expertise of the system depends on the correspondence between the "knowledge-base" and the domain of application. Thus, success in developing these systems depends on a significant investment in task analysis and knowledge elicitation to help discover the distinctions that make a difference. There is little that can be done in the design of either computational engines or graphical representations that can compensate for a failure to understand the domain of application. Unfortunately, the issue of correspondence has often been overlooked in research programs on graphical displays. This is because the correspondence problem demands a specific solution for each particular domain. However, in the search for "general" solutions that apply across domains the basic researcher has focused almost exclusively on issues of coherence. Correspondence has been neglected.

COHERENCE

Coherence refers to the integration of information so that the operator has the ability to "pick-up" the meanings that are represented. One approach is to accomplish this integration computationally. An expert system or automatic assistant integrates the information using a computational model of the work domain and then "filters" the information based on a model of both the work domain and the human operator. "Advice" is provided at appropriate times in support of problem solving; this advice is usually provided whether or not the operator asks for it. Roth, Bennett, and Woods (1987) summarize potential problems with the standard approach to the design of these systems. They found that "... the human's actual role is to amplify the machine's ability to cope with the unanticipated variety in the world or in the problem solving process" (p. 502).

An alternative approach is to integrate the information graphically (or more generally "perceptibly"). The goal is to design the interface so that the constraints are "directly" available to be perceived by the human. To accomplish this, we must have an understanding of human perceptual skill so that we can leverage these skills against the problem of complexity. One of the most powerful perceptual skills that we possess is the capability to extract visual information and recognize patterns in the world. Representational aiding exploits this capability. Alpha-numeric displays can provide very useful information: text can provide detailed messages and warnings, digital displays can provide precise measurements. However, analog displays have the additional advantage of showing a variable, or a set of variables, in context. The current value of a variable can be displayed relative to a boundary or to other variables, and approximate estimates of differences are obtained easily. In situations where several variables testify about higher-level issues configural displays can be used to collect and integrate data into geometrical patterns. Information about the relationships between variables is provided by the spatial interactions in the geometrical pattern. Thus, analog graphical displays transform cognitive activities that strain limited resources (knowing which variables to consider, obtaining each variable, maintaining the values in STM, and mentally integrating them) into perceptual activities (pattern recognition) that are accomplished immediately and effortlessly.

One important use of analogical displays is to preserve the spatial relationships that are inherent in a work domain. Sometimes this will prove difficult. For example, problems will occur whenever the mapping involves converting three dimensional spaces onto two dimensional representations (e.g., topographical map displays to represent terrain). Another issue concerns the determination of those spatial dimensions in the work domain that need to be maintained or emphasized in the representation (as opposed to those dimensions that can be de-emphasized or eliminated). For example, in some cases distance should be appropriately scaled (e.g., for planning a city layout). At other times distance and scaling is secondary to sequencing (e.g., in a route map critical information may correspond to a right turn at the second intersection). Still again, at other times connectedness may be the critical dimension (e.g., is this switch connected to a particular circuit or not?).

In some systems the meaningful properties of a work domain will not be intrinsically spatial. In these cases abstract geometries can be used to illustrate the functional constraints. Process control provides a good case study: many critical properties are not spatial (e.g., mass balance and energy balance are derived measures that have no physical correlates). Perhaps the most well-known example of configural displays is the polar graphic, where an octagonal geometric form is used to represent the overall health of a nuclear power plant (Woods, Wise, and Hanes, 1981). A break in the functional constraints of the work domain (a fault) is represented by a break in the geometrical properties of the display. In the case of the polar graphic display a symmetrical octagon represents a perfectly healthy plant while deviations from symmetry in the display represent faults or developing faults.

Thus, understanding organizational factors in the perception of spatial representations (symmetry, parallelism, closure, etc. -- the classic concerns of Gestalt Psychology) is important. For example, the success of configural displays is dependent upon the highly salient perceptual features that emerge from the interaction of the graphical elements (e.g., the angle that is formed by the meeting of two lines). A difficult, but crucial, aspect of successful display design is to ensure that the salient emergent features produced by these displays correspond to critical functional properties of the work domain (Bennett and Flach, 1992). Once again, it is the mapping from domain to representation that is crucial.

In complex domains the constraints will be numerous, tightly coupled, hierarchically structured, and associated with various levels of importance. The core problem in implementing effective displays is to provide visual representations that are perceived as accurate reflections of this complex and hierarchical structure. Two aspects of this problem are critical. Those aspects of the work domain that are most important should have the most visually prominent, or salient, encodings in the display. The most obvious encoding schemes that can be used to stratify levels of perceptual salience include luminance contrast, chromatic contrast, and emergent features. The second aspect concerns the "nested" structure of constraints in work domains. There may be multiple constraints that are organized hierarchically within a particular level of abstraction in a work domain. This requires multiple, nested geometric symmetries in the representation. The difficulties are exacerbated when the nesting of constraints occurs across different levels of abstraction. For example, the polar graphic represents a set of functional constraints at a very high level of abstraction: overall plant health. However, the physical systems that underlie this high level function are also associated with a set of constraints that are not represented. Designing single representations that incorporate both physical and functional constraints is truly a challenge.

CONCLUSIONS

In theoretical discussions of display design, it is easy to separate coherence (form) and correspondence (meaning); in terms of evaluating the functionality of displays the two are intertwined. Ultimately, form must reflect meaning. In basic research this is often ignored: to say that a particular graphical format (e.g., a triangle display) is good (or bad) ignores the role of meaning. The tasks to be performed are often defined in terms of the visual form that is present on the screen, when the tasks should be defined in terms of the domain semantics that the graphical display represents. To consider meaning and form together is critical to ensure generalizability of findings, and thus represents a major challenge for future research.

REFERENCES

Bennett, K. B., and Flach, J. M. (1992). Graphical displays: Implications for divided attention, focused attention, and problem solving. Human Factors, 34(5), 513-533.

Brown, J. S. (1986). From cognitive to social ergonomics and beyond. In D. A. Norman and S. W. Draper (Eds.), User Centered System Design, (pp. 457-486). Hillsdale, N. J.: Lawrence Earlbaum Associates.

Perrow, C. (1984). Normal accidents. New York, NY: Basic Books.

Rasmussen, J. (1986). Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering. New York, NY: North Holland.

Rasmussen, J., and Vicente, K. (1989). Coping with human errors through system design: Implications for ecological interface design. International Journal of Man-Machine Studies, 31, 517-534.

Roth, E. M., Bennett, K.B., and Woods, D. D. (1987). Human interaction with an 'intelligent' machine. International Journal of Man-Machine Studies, 27, 479-526.

Woods, D. D., and Roth, E. M. (1988). Cognitive systems engineering. In M. Helander (Ed.), Handbook of human-computer interaction (pp. 1-41). Amsterdam: Elsevier Science Publishers B. V. (North-Holland).

Woods, D. D., Wise, J., and Hanes, L. (1981). An evaluation of nuclear power plant safety parameter display systems. In Proceedings of the Human Factors Society 25th Annual Meeting (pp.110-114). Santa Monica, CA: Human Factors Society.