THE INTERNATIONAL JOURNAL OF AVIATION PSYCHOLOGY, *9*(3), 225–240

Functional Interface Design for the Modern Aircraft Cockpit

Gavan Lintern, Tyler Waite, and Donald A. Talleur
Institute of Aviation
University of Illinois at Urbana–Champaign

There is an emerging concern that modern glass cockpits induce information overload. This is sometimes thought to be an inevitable result of the increased complexity and the need for automation that accompanies the transition to high technology. We argue here that the human performance problems created by glass cockpits are not an inevitable consequence of increased hardware complexity or of automation but, instead, are a result of nonfunctional design that increases complexity at the cockpit interface. The essential danger with computerized interfaces is that many physical design constraints are removed and designers are permitted unheralded opportunities for new information and control formats. Low technology forces the use of functional properties at the interface, but computer technology does not. On the other hand, computer technology does not preclude functional design. Computer technology may offer far broader opportunities for functional design by releasing designers from many physical constraints. In this article, we explain the concept of functional interface design and outline how it might enable the use of high technology and automation in the service of robust and cognitively economical action in an aircraft cockpit.

Remarkable developments in computer hardware have encouraged equally remarkable developments in design of the modern commercial cockpit. The use of computers eliminates many of the physical constraints that once shackled interface design. There are possibilities for more extensive automation and for new display and control formats. Although there are many advantages of cockpit computerization, the increased levels of automation and the change in amount and format of information provided to the pilot have been implicated in a series of accidents and incidents. More generally, modern glass cockpits induce special types of pilot errors

---

Requests for reprints should be sent to Gavan Lintern, Air Operations Division, Defense Science & Technology Organisation P.O. Box 4331, Melbourne, Victoria 3001 Australia. E-mail: gavan.lintern@dsto.defence.gov.au

(Sarter & Woods, 1994). To some, it may seem that the automatic functions and information flows found in the modern glass cockpits are taxing the capabilities of many pilots.

In this article, we argue that complexity and information overload are not inevitable consequences of computerization. The problem is rather a consequence of designers being released from many of the physical constraints that have previously imposed certain approaches to interface design. Given their new freedom, designers have sought to provide more automation and more information and have neglected functionality. However, the inherent flexibility of computerization may permit the design of interfaces that exploit principles of functionality even more effectively than is possible with noncomputerized interfaces. To do that will, however, require a deep understanding of the notion of functionality. In this article, we explain functionality as it might be applied to interface design in a modern aircraft cockpit.

The issues we discuss in this article are a subset of those discussed by Rasmussen, Pejtersen, and Goodstein (1994), who provided a comprehensive and detailed discussion of how to move from work domain analysis to implementation in a process of designing a functional interface. Our goal for this article is to motivate an interest in and an understanding of the basis of functional interface design. To that end, our argument takes a different course through a more restricted set of concepts with the purpose of illustrating specific implications for aviation. Nevertheless, any serious attempt to implement these ideas will require attention to the more comprehensive treatment of methods and concepts as outlined by Rasmussen et al. (1994).

A THEORY OF FUNCTIONAL INTERFACE DESIGN

Functionality is about achieving goals. Functional information specifies whether goals can be achieved and how they might be achieved. Functional action supports achievement of those goals. Thus, nonfunctional properties are those that are irrelevant to goals. One task of functional design is to identify functional properties. The notion of functional interface design encompasses a range of ideas identified as direct manipulation, ecological interface design, direct perception, representational design, and semantic mapping (Flach, 1996; Hutchins, Hollan, & Norman, 1986; Vicente & Rasmussen, 1992). One common theme in these concepts is that adult humans are typically very good at recognizing the functional implications of natural information and are also very good at recognizing the functional potential of natural objects and events (Gibson, 1979).

The general design notion is that an operator at a human–machine interface should be able to interact directly with functional properties. These properties will be specific to task-relevant information and objects that can be manipulated. The basis for this approach to design is the belief that our perception–action capabilities are powerfully adaptive so that, by the time of adulthood, extensive experience has prepared us for robust and effortless appreciation and manipulation of a wide range of natural phenomena and objects (Rasmussen et al., 1994). We are genetically endowed to interact with certain natural events, but genetic endowment is insufficient. Our recognition and manipulation of some events becomes robust and effortless through our pervasive experience with them.

In most technological systems, it is neither feasible nor desirable to have an operator directly observe processes or physically grasp and move objects (Figure 1). However, it may be possible to offer an interface in which operators can perceive objects and relations in a representational format that corresponds more closely to physical and temporal spaces and to permit them to manipulate those representations directly. This suggestion should not be taken to imply that an actual object would move in real time as the operator manipulated the represented object. A well-designed direct interface would, however, permit the operator to see current and commanded trajectories in a manner that would make efficiencies and conflicts as evident as if the controller did have a global view of the space and the objects it contained.

Nested and Overlapping Functionalities

We should not take an overly simplistic, unidimensional, or unilevel view of functionality. As noted by Vicente and Rasmussen (1992), functional needs appear at

FIGURE 1 Modern technological systems do not permit direct perception and direct manipulation of objects and processes that need to be controlled. The challenge to the designer is to capture the essential properties of the task in a direct representational format. (This "Bizarro" cartoon by Dan Piraro is reprinted by permission of Chronicle Features, San Francisco. All rights reserved.)

diverse temporal and spatial scales and are nested within and overlap each other. In the natural world, we move seamlessly between scales as the situation demands, and we concurrently attend to multiple and diverse functionalities. Control of an aircraft requires no less. Pilots must be aware of simultaneous needs for control, guidance, and navigation, and within each of these task categories, they must be aware of multiple and sometimes competing requirements.

For dynamical environments, Vicente and Rasmussen (1992) proposed to exploit an *abstraction hierarchy* (AH), which is a multilevel representation of a system at different scales of analysis. One critical feature of an AH is that the different levels are linked by causal (i.e., means–end) relations. Events at one level lead to satisfaction of goals at the level immediately above (Figure 2). A fundamental implication of this model for design is that an adjustment of a property at a lower level can have multiple effects at higher levels. An attempt to satisfy one high-level goal via adjustment of one or more low-level events can have an unexpected impact on other high-level events. This requires either decoupling of the undesired high-level effects from the low-level events or a display that will reveal to the operator the diverse effects of low-level control actions on the various high-level events.

Vicente and his associates (Christoffersen, Hunter, & Vicente, 1996; Vicente, Christoffersen, & Hunter, 1996) have implemented abstraction hierarchies in experimental analogues of process control systems. Dinadis and Vicente (1999/this

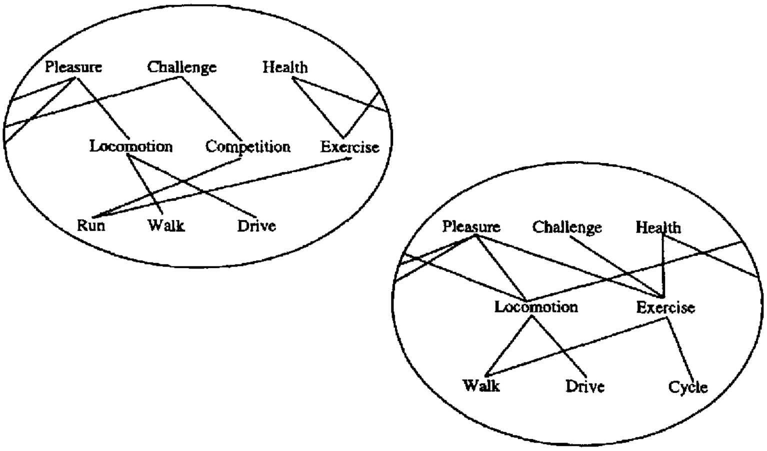

FIGURE 2 A sample of the functional structures of natural activity for two individuals. As shown here, a functional structure is hierarchical and has a means–end relation between levels (the lower levels provide the means to achieving the ends represented at the higher levels). The functional content differs between individuals, and the many-to-many mappings between levels differ between individuals even for shared functions.

issue) demonstrated the relevance of this idea to aviation. The implication for interface design is that the effects specified by an AH must be visible to the operator. It is not that an operator will need to attend to all of these properties at any one time but that he must be able to direct attention to different ones as demanded by the changing situation (Rasmussen et al., 1994).[1]

Direct Perception and Direct Manipulation

The AH presents a schematic outline of functional requirements for a man–machine system but does not, in itself, offer a guide to making these accessible to the operator. Given a comprehensive AH, it would be possible to have the operator derive functional properties from other properties. It is, however, a central proposal of functional interface design that the operator should not be required to derive functional properties but instead should be able to contact them via the robust processes of direct perception and direct manipulation. The combination of nonfunctional properties by mediation or computation to create a functional property is indirect. Assessment of fuel range (a functional property) from a computation involving quantity of fuel, weight of aircraft, and wind velocity does not conform to the requirements of functional design. Neither does reduction nor increase in altitude by manipulation of pitch and power. Thus, there is a need to develop a strategy for representing functional requirements in ways that do not involve operators in calculations, transformations, or inferences that use other properties.

Possibly the clearest explication of directness is offered by Cook (1996) who distinguished between three forms of workstation: *verite, abstraction,* and *ordinateur.* These words might be taken to imply true, remote, and virtual, respectively. The exemplary verite system is a steam locomotive in which controls affect the physical process directly, controls are also displays, and displays that are not an integral part of a control are at least an extension of the physical structure. For example, the throttle is a display of throttle setting as well as a control. A jammed throttle will not open, and the information about the control failure is perceived directly from the thwarted action. When displays of important parameters are separated from their controls, they are nevertheless represented directly. For example, the display of water level in the boiler is a physical extension of the boiler itself. At a higher level of the AH, the effect of opening a throttle may be displayed on a meter, but the engineer

---

[1] The notion of a multilevel, functional hierarchy, which is the central idea of AH, has been seriously neglected in human factors research and even in variants of functional interface design in which consideration of an isolated functionality at one level has been the norm. Rasmussen and his associates (in particular, see Rasmussen, Pejtersen, & Goodstein, 1994) have continued to develop the AH as the foundation for their approach to design.

is also assailed directly by information about the effect of this control action in the form of increased vibration, sound, temperature, and speed.

The advent of electrical control permitted development of abstraction systems in which controls could be remote from the process and would no longer work directly on it. The lowering of aircraft landing gear is of such a type. The lever will operate even if the wheels are jammed. A sensor and associated display are required to ensure that the system has worked. However, one strong feature of verite systems is retained in the abstraction system, that being the one-to-one mapping between control and process and between process and display. The introduction of computers has taken us further from the verite system by removing even this design constraint. This results in the ordinateur system in which only a subset of controls and displays are presented to the operator at any one time, and single controls are used for multiple functions.

Cook's (1996) argument should not be taken as a plea to return to the verite system. The plea is rather to understand the nature of verite systems and to develop computerized workfaces that incorporate their more valuable features. This involves adherence to principles of direct perception and direct manipulation.

Representation of Functional Properties

A major challenge to designing interfaces that exploit direct perception and direct manipulation is to develop compelling perceptual representations of spatial, temporal, and relational properties. The belief that this is possible stems in part from the observation that experts often speak of their work as if they can directly perceive (especially visually) objects and relations that are not within their perceptual field.

An example of functional visualization is offered by Rochlin (1991), who described the experience of the Tactical Coordination Officer in the Command and Control Center of a U.S. Navy Fleet as one of "seeing" the three-dimensional space surrounding the fleet. This expert receives information in messages from other operators in the center and builds a "mental picture" of this space and the objects in it. Relations and potential capabilities are an important part of the image (e.g., can my own aircraft at a specified location effectively intercept enemy aircraft before they can cause damage to the fleet?). Tactical Coordination Officers refer to the experience as "being in the bubble" and speak of it as a visual experience. Aviators might describe a similar experience in terms of situational awareness. When operators experience something like this as a visual event, it seems likely that we could represent it visually.

More generally, Rasmussen et al. (1994) argued that designers should examine the work domain (e.g., manuals, displays, operator knowledge) for images that suggest the types of representations that can be employed at a specific information interface. Within aviation, this would require that the designers develop an understanding of how experienced pilots mentally visualize functionalities as a guide to how a computer interface might help satisfy their functional requirements.

Computer-Supported Interfaces

To be direct means that the operator has access to the actual physical properties of the process. In that a computerized interface mediates between the operator and the physical processes of the system to be controlled, a computer-generated representation is necessarily indirect. However, the design aim is to provide an isomorphic mapping between functional properties of the physical process and their representations at the interface (Vicente & Rasmussen, 1990). Functional properties are then uniquely specified at the interface, and although access to them is indirect, the operator can rely on direct processes to perceive and to manipulate their representations. We should remember, however, that the designer selects the properties to be specified at the interface. Although the design goal is to omit irrelevant properties and to accentuate relevant properties, there is always the danger that pursuit of this goal will not be entirely successful. Development of a comprehensive AH is the best means of ensuring that all relevant functional properties are represented at the interface.

Direct is sometimes taken to mean "manual" as in "nonautomatic." The problems of automation are, however, not with automation in itself but with many of its implementations. There is an attempt with some automatic systems to minimize the involvement of the operator, a strategy fraught with risk in an unpredictable and tightly coupled environment (Perrow, 1984). In other systems, the results of automatic processes are returned as nonfunctional properties that involve the operator in further computations. The goal of direct design is to return the results of computations in the form of functional properties that do not involve the operator in further computations.

A Common Input–Output Language

Hutchins et al. (1986) argued that a direct interface is one in which the output properties are written in the language of the input properties. To illustrate, one might ensure successful arrival at a destination by pointing toward that destination rather than through setting a course by compass reference. One might also adjust comfort related to environmental temperature and humidity by adjusting a comfort index in preference to setting a temperature. Representations can thus be seen as the properties they refer to. It is this sort of direct interface that will support robust control behavior. As is evident in the notion of a comfort index, the development of a common language is not necessarily easy. This remains one of the challenges for those working in this area.

It is essential that any common input–output language be of natural and pervasively experienced physical relations (Figure 3). We do, of course, develop considerable awareness of certain numerical scales that represent functionally important physical magnitudes. Appreciation of the comfort level of certain ranges of environmental temperature is one. Here, system input (temperature) does not share a common language with system output (comfort). There is an inevitable need for an inferential step, and it is a basic assumption of functional interface design that the inferential step retards development of robust and flexible appreciation of the input–output relations. Of particular concern is that other physical magnitudes may influence the relevant functional property (wind chill on comfort in the aforementioned example) in ways not fully anticipated by even an experienced person.

Transforming the Task

Hutchins (1995a) criticized the predominant attitude in design of computerized interfaces towards development of intelligent agents that relegate the role of an operator to that of monitor for normal operations and last line of defense in emergency. The automatic pilot and the modern day flight-management computer might be seen as examples of such intelligent agents. Following an ethnographic analysis of shipboard navigation, Hutchins (1995a) argued for representations that transform tasks to forms that exercise the powerful abilities of the human operator to match patterns, to manipulate simple physical systems, and to anticipate the action of simple physical systems.

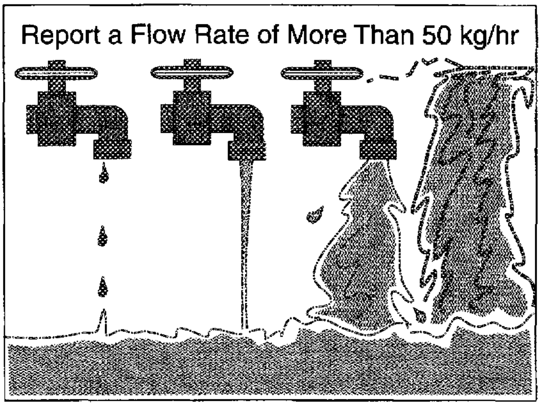

FIGURE 3 In one nuclear power plant, operators are required to recognize a fluid leakage of more than 50 kg/hr (Vicente & Burns, 1996). What does that mean? Is it a slow drip, a steady flow, a gush, or a torrent?

An aviation example can be used to illustrate what is meant by task-transforming properties (Hutchins, 1995b). The speed bugs on the air speed indicator in many commercial aircraft help pilots coordinate flap settings with airspeeds. These speed bugs replace memory and computational requirements with pattern matching requirements. The command speed bug is particularly important. One common requirement during a landing approach is to monitor whether actual speed is within 5 kt of commanded speed. At first glance, this might seem to be a problem in mental arithmetic. However, the base of the speed bug is 10 kt wide (apparently for reasons unrelated to the task of maintaining speed within 5 kt of commanded speed) so that pilots merely have to ascertain whether the airspeed needle is pointing at any part of the speed bug. "This strategy permits a conceptual task to be implemented by perceptual processes" (Hutchins, 1995b, p. 283).

Task transformations that lead to use of perceptually scaled relations are in accordance with the concept of functionality as previously outlined. The potential to implement powerful transformations of tasks has become a reality of computerized interfaces. One design goal should be to ensure that the task transformations we implement lead to more robust and cognitively economical activity. Our experience with computerized interfaces shows that just the opposite has occurred (Cook, 1996; Sarter & Woods, 1994). We might ensure that task transformations, as implemented in computerized interfaces, lead to robust and cognitively economical activity by adhering to the principles of direct perception and direct manipulation.

Affordances

Some developments of the AH (Vicente & Rasmussen, 1990) have been heavily influenced by insights drawn from Gibson's (1979) views on the structure of natural activity. In particular, the notion of multiple, overlapping and hierarchical functionalities follows Gibson's views of the nature of the affordance structure (cf. Figure 2). His notion of affordance offers a succinct statement on functionality. In the natural world, the affordance for crossing a pond on stepping-stones is the ratio of the maximum distance between stones and the maximum step distance. Similarly, whether or not the situation of an approaching missile affords safe evasion depends on time elapsing before arrival (judged perceptually) and time required to execute the evasion. In both examples, the affordances are ratios of identically dimensioned quantities and are therefore dimensionless numbers. More generally, dimensionless criteria are sometimes referred to as pi numbers[2] (Buckingham, 1914).

---

[2] In high school geometry we learn that pi equals 3.14159265. This is a dimensionless number. However, the concept of a pi number is far more general than implied by the usual high school introduction.

Functionality and Dynamics

To place this in a flight-related context, Stanard, Flach, Smith, and Warren (1996) simulated a low-level flight scenario to examine informational properties involved in collision avoidance. By the use of two different dynamic responses in their simulated aircraft, they were able to demonstrate a relation between essential information and characteristics of the dynamic response. The task was to initiate a climb at the last possible moment to avoid collision with an obstacle. Forward speed was varied across trials in both conditions. Some participants were given a simulation in which the climb rate was constant, and others were given a simulation in which the climb rate was proportional to forward speed. The optimum strategy for a constant rate of climb was to initiate climb at a fixed time from the obstacle. The optimum strategy for a rate of climb proportional to forward velocity was to initiate climb at a fixed distance from the obstacle. The data show that participants used the strategy that was best for their condition.

Information for time-to-contact and information for distance were both available to the participants. This experiment revealed the reciprocity between information and dynamics. In other words, the dynamic response of the controlled system determines the nature of the information required for control. This suggests that, in natural environments, we need to ensure that control of a dynamic response does not require informational properties not readily available, and in designed environments, we need to ensure that informational properties required for control of a dynamic response are clearly displayed. To be consistent with the previous discussion on affordances, this information needs to be presented in a manner that enables the operator to apprehend the relation between the essential control information and the dynamic response of the system.

Summary

The key features of functional interface design are encapsulated in the ideas of multiple, nested, and overlapping functionalities, direct perception and direct manipulation, and dimensional identity of input and output properties. There can be little doubt that the cockpit of any new aircraft will be computerized and will have many automatic capabilities. The major challenge is to constrain the almost infinite number of possibilities enabled by computer technology to those that conform to the perceptual–cognitive–active capabilities of the general population from which our pilots will emerge. This will ensure that the required piloting tasks are robust and are economical of cognitive effort.

FUNCTIONAL CONTROL, GUIDANCE, AND NAVIGATION

The Flight Management Computer

The challenge to coordination of activities and the deficiencies of the flight management computer (FMC) in supporting the task of piloting an aircraft is illustrated in Casner's (1994) observation on crew–system coordination on the flight deck of a commercial aircraft. He observed that pilots like to use the automatic functions of the FMC in guidance and navigation for predictable but not for unpredictable situations. Predictable events allow pilots to complete the FMC setup ahead of time, but setup for unanticipated changes can take so long that pilots prefer to revert to manual control.

Hutchins (1995a) argued that, for spatially and temporally extended tasks, much of the challenge is in the coordination of various subtasks. A workspace should be designed to support that coordination. This is, however, one area in which modern flight management computers appear to be deficient (Sarter & Woods, 1994). Most of the automatic modes of the FMC are, in Hutchins' terms, intelligent agents rather than task transforming representations. The issue of most concern in modern cockpits is that the nature of the task has been changed from one that is robust and cognitively economical to one that is brittle and, in many situations, cognitively intensive.

Object Representation

For operators to revert to manual control in times of high workload is a glaring indictment of any modern computerized system. Casner (1994) noted the preference of pilots to preconfigure their system. In doing so, they appear to be adhering to the design principle of representing functional properties (their course and its waypoints) as objects. However, the system does not permit preconfigurations that could help pilots deal with unanticipated course changes. From the perspective introduced here, functional properties (e.g., waypoints) should be represented as objects that pilots can manipulate to configure a new course. If the task of developing object representations as functional properties is to be left to pilots, the interface must be designed so that these object representations can be configured prior to takeoff.

Development of object representations from nonfunctional properties will generally be cognitively intensive and prone to error. The issue here is whether that is done by the designer or the pilot and, if by the pilot, whether it is done prior to takeoff or in flight. Only an appropriate work-domain analysis (Rasmussen et al., 1994), which is beyond the scope of this article, could ascertain whether a library of object representations developed by the designer could be sufficiently flexible. However, this task cannot be left to the pilot to complete during flight.

Intuitive Pictorial Displays

Oliver (1990) argued that situational awareness is compromised in regard to aircraft systems and in regard to the flight situation by many of the display formats currently used in cockpits of modern commercial aircraft. Fuel state is one element of systems information that must be available to the pilot. Oliver (1990) suggested that a classic piping and instrumentation diagram would support situational awareness more effectively than standard meters. Dinadis and Vicente (1999/this issue) took a systematic approach, based on the strategy outlined by Rasmussen et al. (1994), to redesign the flight engineers station in a C130 transport. They used a diverse array of pictorial forms to represent both fuel and energy states at different levels of the AH. Although no performance evaluation of this new interface has yet been undertaken, C130 flight engineers were generally positive in their evaluation of the new design.

The new-style navigation displays already in wide use offer more effective support than the conventional horizontal situation display for awareness of the flight situation by depicting aircraft position relative to airports, navigational aids, and waypoints (Oliver, 1990). However, they depict very little of the terrain topography, and setup remains problematic. Both of these factors have been implicated in the crash of American Airlines Flight 965 in December of 1995 (Aeronautica Civil of the Republic of Colombia, 1995). By neglecting the multilevel nature of complex action, designers may have developed pictorial formats that enhance situational awareness on some important dimensions but degrade it on others.

Situational awareness can be both spatial and temporal. It can expand and shrink on both dimensions as circumstances change, and it is based on an appreciation of functional–physical relations. In natural activity, the transitions between spatial and temporal scales and the redirection of attention to different relations are generally automatic and seamless. There is no need for conscious selection. Information is available and is noticed as the situation demands. This notion represents a serious challenge for those who must design the representation of multidimensional and hierarchical information in such a manner that it does not interfere when not needed but is immediately available when the situation requires attention to it.

Direct Manipulation

Direct manipulation offers one means of transforming tasks into robust and cognitively economical forms. To illustrate, wheels might be lowered by movement of a suitable icon. In abstraction systems, as described by Cook (1996), the response of the controlled object is represented in a dedicated display (e.g., wheels or flap status indicators). In verite systems, the control is also the display (we grasp the wheel or flaps and move them directly; Figure 4). To return to the principle of a verite system in a computerized cockpit, the pilot would apply pressure to the icon as control, which would move as the wheels actually move. In the case of wheels stuck in the "up" position, the pilot would not be able to move the control. It would, however, be essential to ensure that lack of resistance in a control, which would normally signal correct operation, is not the result of a failure in the feedback system. An initial resistance that diminished only as the wheels actually extended could possibly assure this.

To change altitude by adjustment of pitch and power is indirect. For direct manipulation, an object representing the aircraft would be moved to the new altitude. The flight director is an example of a direct manipulation system that already exists in aviation. With this system, a new altitude is set and a cursor shows the pilot how the aircraft is to be controlled to achieve that altitude. Note that an autopilot, which has the additional feature of automatically moving the aircraft to the new altitude, eliminates the possibility of direct manipulation and thus lacks one central feature of the verite system, that being the control as the display. In general, verite principles suggest that automatic prediction is useful but that automatic control is not.

Comment

For the central tasks of control, guidance, and navigation, there are a number of possibilities for implementing functional design at various levels of the AH. Some sys-

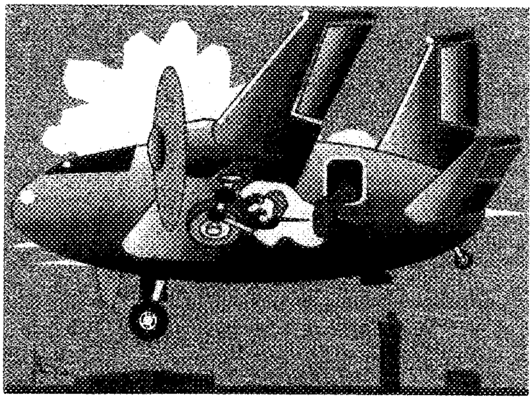

FIGURE 4 The most primitive form of direct manipulation. The issue for functional interface design is whether we can capture the essential directness of this action without forcing the operator into a physically unworkable and dangerous manipulation.

tems employed in aviation already correspond to the functional perspective, but other systems do not. There is a crucial need for a more systematic approach to interface design for the modern aircraft cockpit that retains and builds on the strengths of current designs as it eliminates their weaknesses.

CONCLUDING COMMENT

Modern aircraft cockpits are so complex that they tax the capacity of pilots to process and comprehend the information provided. One prevailing view is that this is an inevitable result of the progress of technology and that the answer lies in better training tailored specifically to counter information-processing limitations of the pilot. Without negating the essential role of training, the perspective taken from functional interface design is that this is not primarily a training problem. Training should not fill the role of compensating for poor design but should be oriented toward facilitating the development of expertise with well-designed systems. Thus, the first requirement is to design a functional interface that supports robust and cognitively economical behavior.

The evolution of procedures (or practices) of work is shaped by cultural and technical context. For example, specific seagoing navigational techniques developed within a culture have been influenced by the needs, the base of knowledge, and the current inventory of technical artifacts that exist within that culture (Hutchins, 1995a; see also Lintern, 1996). For aviation, there may be some impetus to break with patterns of technical development that have guided aircraft and cockpit design over decades and to transition to newer, more efficient patterns. Nevertheless, we must recognize that new designs may be more efficient for designers and manufacturers but not for pilots. We are now in an era in which the rapid pace of technological advancement encourages a revolutionary versus evolutionary approach to design. In following this path, we must ensure that we do not eliminate the functional cockpit features and practices that have evolved over decades unless we replace them with equally functional alternatives.

The fundamental argument outlined in this article is that the reduction of functionality in the modern cockpit is not an inevitable result of computerization and that it is possible to design a radically new cockpit interface that would fit the pervasive and robust capabilities of most normal adults. A cockpit that permits a pilot's direct interaction with important functional properties will make the art and skill of flying easier to acquire and will reduce problems of relearning, refamiliarization, and type transition. Most generally, a primary design goal should be to relieve pilots of in-flight computations and inferences while leaving them in full control to the extent that they can adapt to unanticipated changes or events. The functional approach would enable this by providing pilots with the means to precompute relations, to preconfigure models as objects, and to develop low-effort strategies to accomplish complex tasks.

One criticism occasionally directed at variants of functional interface design is that the principles and implementations are obvious, and many of them have been proposed from other design perspectives. On the one hand, we should expect that principles for design of an "intuitive" interface are themselves intuitive and that some of the implementations are little more than common sense applications of those intuitions. On the other hand, there are many popular design features that do not conform to the fundamentals of functional interface design. The contrast between a flight director and an automatic pilot offers one example, and this specific example illustrates the more widespread concern with implementations of automatic processes. Functional interface design is not about removing automation but about employing it in service of functional action. The primary purposes of this article are to make explicit the reasons for designing interfaces in certain ways, to provide principled arguments for avoiding functionally indirect interface features, and to outline a context for developing a comprehensive interface design that will support all required functions at all desired levels of abstraction.

ACKNOWLEDGMENTS

Gavan Lintern is now at the Defense Science & Technology Organisation, Melbourne, Australia.

Preparation of this article was supported by the Federal Aviation Administration via the National Aeronautics and Space Administration under Contract ATI AGATE 6212. Peter Hwoschinsky was the technical monitor. Kim Vicente, University of Toronto, and Thomas A. Stoffregen, University of Cincinnati, reviewed drafts of this article.

REFERENCES

Aeronautica Civil of the Republic of Colombia. (1995). *Aircraft accident report: Controlled flight into terrain, American Airlines Flight 965, Boeing 757-223, N651AA near Cali, Colombia, December 20, 1995.* Santafe de Bogotá, DC, Colombia: Author.

Buckingham, E. (1914). On physically similar systems. *The Physical Review, 4,* 344–376.

Casner, S. M. (1994). Understanding the determinants of problem-solving behavior in a complex environment. *Human Factors, 36,* 580–596.

Christoffersen, K., Hunter, C. N., & Vicente, K. J. (1996). A longitudinal study of the effects of ecological interface design on skill acquisition. *Human Factors, 3,* 523–541.

Cook, R. I. (1996). *Verite, abstraction, and ordinateur systems in the evolution of complex process control.* Seminar presented at the University of Illinois, Champaign.

Dinadis, N., & Vicente, K. J. (1999/this issue). Designing functional visualizations for aircraft systems status displays. *The International Journal of Aviation Psychology, 9,* 241–269.

Flach, J. M. (1996). Situation awareness: In search of meaning. *Crew System Ergonomics Information Analysis Center Gateway, VII*(1), 1–4.

Gibson, J. J. (1979). *The ecological approach to visual perception.* Boston: Houghton Mifflin.

Hutchins, E. (1995a). *Cognition in the wild.* Cambridge, MA: MIT Press.

Hutchins, E. (1995b). How a cockpit remembers its speeds. *Cognitive Science, 19,* 265–288.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1986). Direct manipulation interfaces. In D. A. Norman & S. W. Draper (Eds.), *User centered system design: New perspectives on human–computer interaction* (pp. 87–124). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Lintern, G. (1996). A review of "Cognition in the wild" by Edwin Hutchins. *The International Journal of Aviation Psychology, 6,* 299–306.

Oliver, J. G. (1990). *Improving situational awareness through the use of intuitive pictorial displays.* Warrendale, PA: SAE International.

Perrow, C. (1984). *Normal accidents: Living with high-risk technologies.* New York: Basic Books.

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

Rochlin, G. I. (1991). Iran air flight 655 and the USS Vincennes: Complex, large-scale military systems and the failure of control. In T. R. La Porte (Ed.), *Social responses to large technical systems* (pp. 99–125). London: Kluwer.

Sarter, N. B., & Woods, D. D. (1994). Pilot interaction with cockpit automation II: Operational experiences with the flight management system. *The International Journal of Aviation Psychology, 4,* 1–28.

Stanard, T., Flach, J. M., Smith, M., & Warren, R. (1996). Visual information use in collision avoidance tasks: The importance of understanding the dynamics of action. In *Third Annual Symposium on Human Interaction With Complex Systems* (pp. 62–67). Los Alamitos, CA: IEEE Computer Society Press.

Vicente, K. J., & Burns, C. M. (1996). Evidence for direct perception from cognition in the wild. *Ecological Psychology, 8,* 269–280.

Vicente, K. J., Christoffersen, K., & Hunter, C. N. (1996). Response to Maddox critique. *Human Factors, 38,* 546–549.

Vicente, K. J., & Rasmussen, J. (1990). The ecology of human–machine systems II: Mediating "direct perception" in complex work domains. *Ecological Psychology, 2,* 207–249.

Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics, 22,* 589–606.

Manuscript first received January 1999