M. Mouloua and J.M.Koonce (Ed.), *Human-automation interaction: Research and practice* (pp. 19-28). Mahwah, N.J.:Lawence Erlbaum, 1997.

DO YOU KNOW WHAT MODE YOU'RE IN?
AN ANALYSIS OF MODE ERROR IN EVERYDAY THINGS

Anthony Andre
Interface Analysis Associates
San Jose, CA

Asaf Degani
San Jose State University
San Jose, CA

ABSTRACT

Much due attention has been recently devoted to the issue of human interaction with reactive, complex systems, such as the aircraft cockpit. For example, we have learned from several case studies and empirical investigations that increased levels of automation is not a panacea, often due to the inability of the interface to effectively communicate the status and behavior of the various modes of the machine. Yet, such problems are not unique to the aviation domain, nor to complex systems in general, and instead, play an important role in our interaction with less complex everyday products. From remote controls and VCRs to telephones and automotive cruise controls—mode errors are commonplace. In this paper we describe a framework for representing modal systems and show how this framework can detect potential mode problems, using an automotive cruise control system as an example.

INTRODUCTION

We in the human factors community are well aware of problems attributed to automation in the modern commercial aircraft. Researchers have identified "lack of mode awareness" as one of the most critical of these problems (e.g., Sarter and Woods, 1995a). Mode awareness refers to the operator's knowledge and understanding of the current and future status and behavior of the system, given a particular mode.

The errors and related problems that result from a reduction in, or loss of, mode awareness are sometimes referred to as "automation surprises"—an ill-defined yet commonly used phrase (Sarter and Woods, 1995a). The presence and consequences of mode errors in the context of human interaction with automated flight control systems have been surveyed (James, McClumpha, Green, Wilson, and Belyavin, 1991; Sarter and Woods, 1992; Wiener, 1989), analyzed from incident reports (Eldredge, Dodd, and Mangold, 1991; Vakil, Hansman, Midkiff, and Vaneck, 1995), modeled (Callantine and Mitchell, 1994; Degani, Mitchell, and Chappell, 1995) observed and documented during actual flights (Degani, Shafto, and Kirlik, 1995; Wiener, 1989), empirically studied (Sarter and Woods, 1994, 1995b), and summated (Billings, 1996).

Indeed, mode errors have been directly linked to several recent accidents involving modern transport aircraft (*Aviation Week and Space Technology*, 1995).

A review of this literature on mode errors might lead one to conclude that the phenomenon is exclusive to the aircraft cockpit, or other complex, automated systems. Yet, mode errors are not confined to the aviation domain, nor to complex systems in general, and instead, play an important role in our interaction with many simple, less automated, everyday consumer products and systems. From remote controls and VCRs to telephones and automotive cruise controls—as products become more digitally based, allowing a higher number of functions—the number of modes (states) per given product increases. As a consequence, the potential for, and magnitude of, mode errors induced by these products is correspondingly higher.

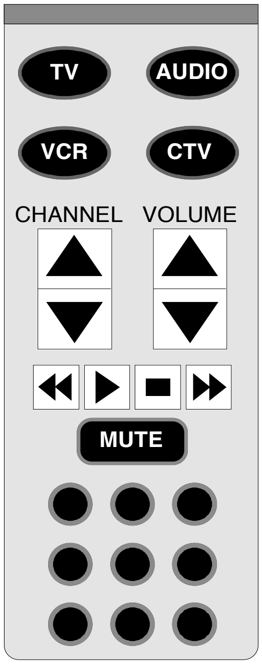

Figure 1. Remote Control

Take, for example, the remote control illustrated in Figure 1. This is a typical "universal" remote control that can operate any of four entertainment systems: the TV, the stereo (audio), the VCR, and the cable TV box. Alas, this four-fold functionality is not a panacea. Imagine that you want to watch a movie on your VCR. You press the "VCR" button, then press play to start the tape. While watching the movie you get an important phone call and rush to mute the TV volume using the remote. You press the "mute" button, but nothing happens. Why?

Since you are still in the "VCR" mode, the remote control does not respond to "TV" commands, "mute" being one of them. In order to mute the volume, you must first press the "TV" mode button and then press the "mute" button. But how does one determine which mode they are currently in? The remote doesn't provide the answer since the buttons do not change position to reflect their state, nor is there a display to indicate the current state. Instead, the user has to remember (i.e., keep a log of) which mode button they last pressed. Thus, given the logic and interface design of this remote control, if we assume the user doesn't remember what mode they last pressed (a worst case, but common scenario), the user can not predict the effect of pressing the "mute" button.

WHAT'S IN A MODE?

Modes are implemented, in one way or another, in almost any modern machine or piece of equipment (Monk, 1986). Simply stated, modes represent the different behaviors, or functions, of a given system. Thus, the more functions, the more modes (Norman, 1981). Some products have long been associated with multiple modes; for example, many electronic wrist watches have time, alarm, timer, and chronometer functions, each accessed by a separate control or different combination of control presses. Further, other products that long afforded only one manner of behavior now can operate (behave) in many different, often mutually exclusive, ways. For example, fax machines no longer just fax; they now copy and print as well. Answering machines no longer have a single mail box, but commonly have three. The result of this increase in functions is the mapping of several modes to a lesser number of controls (e.g., dials, buttons, switches, etc.). Thus, instead of a unique set of displays and controls for each mode, the same set of displays and controls carries out different functions depending on the mode selected. The result of this many-to-one mapping is an increased propensity for mode errors, much the same as has been studied and observed in the aircraft cockpit.

Of course, the savvy reader might reasonably suggest that a mode error in the context of a consumer product is not as important as a mode error in the context of a high-risk systems such as commercial aviation, medical, and military. It is our position, however, that the consequence of mode errors should not dictate or constrain the domain in which they are studied. In this paper we argue that one way to learn about common mode errors in human-machine systems is by representing and studying the structure of the system—in particular, the relationship between the machine's internal states and what is presented to the user about these states via the interface. And, that some of these unique machine-interface relationships that produce mode errors are independent of the complexity of the device and the domain in which it is employed. In essence, we are interested in the basic architectural structure of modal systems that induce mode errors. We therefore believe that the proper approach to the study of these systems is one that is both device- and domain-independent.

A MODELING FRAMEWORK

In the following section we present a modeling framework called *Ofan*. This framework was developed for describing human interaction with systems that have modes (Degani and Kirlik, 1995). After describing the framework, we show how the framework can be used to represent modal systems and highlight the (sometimes) hidden states within the human-machine-environment system that may lead to mode errors.

The Ofan Framework

The Ofan framework's language is Statecharts. The theory behind modeling the human in this framework is based on the Operator Function Model (OFM). Both Statecharts and OFM share the same underlying language—the Finite State Machine (FSM)—which provides a medium for describing the control behavior of a given system in terms of its states and transitions (Minsky, 1967).

The Modeling Languages

Statecharts is a specification language for complex, real-time, reactive systems (Harel, 1987, 1988). The formalism allows for hierarchy and concurrency. Hierarchy is represented with the notion of a super-state and sub-states encapsulated within a super-state. Concurrency is provided by the notion of independence and orthogonality such that a super-state containing two orthogonal sub-states can be described as their product (.AND. relationship). Another feature of Statecharts is its broadcasting mechanism—broadcasting the active states and events (that trigger transitions) to the entire network. The Statecharts formalism allows for several other enhancements such as default entries into a state, conditional entries, and selection- and history-based entries.

The Operator Function Model (OFM) is a task decomposition formalism used for task analysis and simulation (Jones, Chu, and Mitchell, 1995; Mitchell and Miller, 1986). Depending upon demands from the environment (e.g., ATC clearance) and the requirements of the task (e.g., mandatory procedures), the OFM specifies the progression of tasks and actions (either manual, perceptual, or cognitive) that a user will execute.

The Ofan framework represents the system as a set of five concurrently active modules: 1) environment, 2) human function and task, 3) controls, 4) displays, and 5) machine. Each module contains a process and broadcasts its active states and events to other modules in the network, often triggering other modules to change states. Through these broadcast communications, the five modules in the framework are all interconnected to form a whole—the human-environment-machine system. The system representation is analogous to a set of cog wheels: an event in one wheel affects the adjacent wheel and so on (and hence the name 'Ofan'—Hebrew for a set of perpetuating wheels).

CRUISE CONTROL: A HUMAN-MACHINE SYSTEM

The Ofan framework and our approach for modeling modes allow us to describe systems that exhibit several levels of human-machine involvement. One such system is a cruise control of a modern car. We first describe an incident involving this system, then present a model of this system, and finally discuss the unique feature of the system's design that contributed to the incident. Due to space limitations, we will only present part of the framework (see Degani and Kirlik, 1995; and Degani, Shafto, and Kirlik, 1996, for a more complete treatment).

A Mode Error

The incident occurred while the second author was driving on a highway during a rainy night. The traffic was slow at about 40 miles per hour. Bored and tired, the driver engaged the cruise control by turning it ON and pressing the "set-speed" button. The cruise control engaged, and the car cruised at 40 mph. Several minutes later the rain stopped and the traffic speed increased; subsequently, the driver depressed the gas pedal to manually override the cruise control and increase the speed to 60 mph. He drove in this configuration for some 10 miles until coming to his planned exit from the highway. At this point he had completely forgotten that the cruise control was previously engaged (there was no indication in this type of car that the cruise control was on and engaged).

The exit ramp was initially sloped downhill and then extended uphill ending with a curve into a busy intersection. Aware of this landscape, the driver planned to release the gas pedal and let the car glide downhill (lifting the foot from the gas pedal) and maintain a slow speed during the turn into the intersection. Initially it all worked as planned. However, once the car reached a speed of just below 40 mph the cruise control "kicked in." Not expecting such a jolt, the driver lost control of the car as it sped into the intersection. Luckily, no other cars were present at this late-night hour.

Representing the Modal System

There are five levels of involvement, in terms of speed control, in this human-machine system. Each level of involvement is defined as a combination of machine and human state-of-affairs, with respect to speed control. The levels range from fully manual to fully automatic (see Table 1). The levels of involvement are represented via hierarchical encapsulation of levels (sub-states) within larger levels (super-states). Figure 2 depicts these relationships.

At level one ❶, the cruise control is off and the car's speed is controlled completely manually (OFF / *manual*). At level two ❷, the cruise control is turned "on," but speed is still controlled manually by the driver. Note, however, that the configuration of the human-machine system has changed: it is cruise-control ON with respect to machine state, and MANUAL with respect to human state (ON / *manual*). In essence, the configuration of the human-machine system is a vector of these two components. Once the driver presses the set-speed button, the current speed (say 40 mph) becomes the set speed, the machine engages, and we jump immediately to level five ❺. At this level the speed is actively controlled by the machine and the human is idle (ON; SPEED SET; ENGAGED; ACTIVE / *idle*).

Table 1. Levels of Involvement.

Level	Machine	Human
❶	OFF	*manual*
❷	ON	*manual*
❸	ON; SPEED SET	*manual*
❹	ON; SPEED SET; ENGAGED	*manual-override*
❺	ON; SPEED SET; ENGAGED; ACTIVE	*idle*

If the driver now decides to increase speed, he or she depresses the gas pedal and manually overrides the automatic speed control, as depicted in level four ❹ (ON; SPEED SET; ENGAGED / *manual-override*). Note that the cruise control is still engaged, but the speed of the vehicle is no longer actively controlled by the machine. Nevertheless, there is a lower bound on the speed (40 mph in the case of aforementioned incident) at which, when current speed is equal to the set speed, a transition to the fully automatic ACTIVE state will occur automatically (depicted as a broken arc in Figure 2). When this happens we are back to level five.

Finally, the transition to level three ❸ will occur if, while the machine is engaged, the brake pedal is depressed. In this case the speed-set value is still stored in the machine's memory and the machine is ON, but the driver is in a completely manually mode (ON; SPEED SET / *manual*). Once the "resume" button is pressed, the machine will be re-ENGAGED and transition to the fully automatic ACTIVE state again (level five).

These five levels represent the continuum, in discrete and hierarchical steps, from fully manual control to fully automatic control, with respect to the speed control of the car. The hierarchical structure of the system and the Statecharts formalism provides such that an exit from any super-state causes an exit from every one of its sub-states. For example, notice that turning the cruise control "off" causes an immediate transition into fully manual, regardless of which level we were previously at. This hierarchical approach for describing such human-machine systems is quite different from the common engineering approach for modeling similar systems (cf. Smith and Gerhart, 1988).

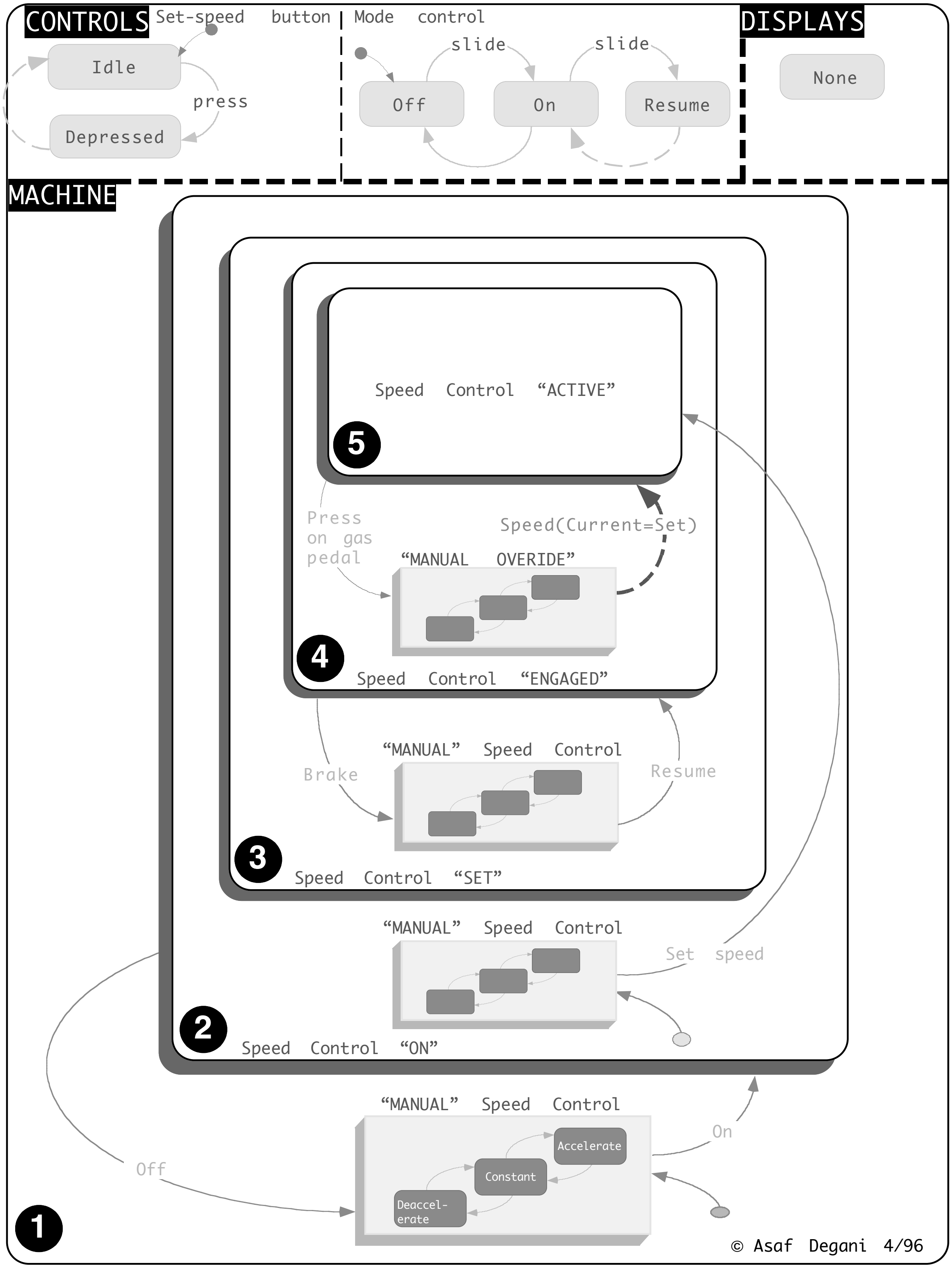

Figure 2. A representation of the system's states and transitions.

Analysis of the Mode Error

Generally speaking, in order to anticipate the next behavior of a machine, one requires three sets of information: One, a complete and accurate model of the system in terms of states and transitions. Two, sensing of both internal and external stimuli (that fire transitions between states). And three, the ability to keep track of (or log) the system's states. In this analysis we impose a constraint that the ability to keep track of the system's states can, and should be done, by reference to feedback provided via a display; no memory of past behavior should be required.

In attempting to understand the mode confusion that was described in the incident, we focus on the automatic transition from ENGAGED / *manually override* (level four) to ACTIVE / *idle* (level five) after releasing the gas pedal and reaching the set speed. This is the only automatic transition in this system; all other transitions are initiated by neuro-motor actions (e.g., depressing the brake, pressing the "on" button). With the aforementioned requirements and assuming the driver has no memory of the last transition, we find that this human-machine system is non-deterministic. That is, we can't resolve between two different transitions that may occur after the driver releases the gas pedal: 1) the cruise control system engages and activates when the car's speed reaches the set speed (40 MPH), or 2) the cruise control system never engages and the car's speed continues to decline (as was expected by the driver). Note that the position of the control switches does not aid the driver in resolving this ambiguity. Moreover, this holds true even if he or she has an absolutely complete and accurate model of the system! This is because the settings of the cruise control interface are the same regardless of whether the system is at level two (ON / *manual*) or level four (ON; SPEED SET; ENGAGED / *manual-override*).

SUMMARY

Issues related to mode awareness and mode errors have received much attention in the aviation literature. We show that modes are not unique to the aircraft cockpit and in fact, permeate everyday things—from remote controls to automotive cruise controls. We argue that the architecture of mode systems that lead to mode errors can be revealed by representing the structure of the system—in particular the relationship between the machine's internal states and what is presented to the user about these states via the control/display interface. We illustrate this point using a framework, called Ofan, which depicts the system's states and transitions, and the overall structure of human-machine involvement. Further, we believe that his framework not only allows behavioral scientists to better understand the inherent structure of modal systems, but perhaps more importantly, allows engineers/designers and human factors specialists alike to evaluate modal systems using a common language.

Our message to designers and purchasers of modal products and systems is clear and simple: An increase in functions (modes) does not necessarily result in an increase in functionality; indeed, it can actually result in a decrease. For the user to take advantage of the various modes inherent in a given product, they must be provided an interface that, at a minimum, provides clear feedback regarding the status (mode) of the system. After all, what's the purpose of providing multiple modes if the user can't effectively distinguish between them? It is our further belief that the technique employed here can be used to reveal structures of both simple and complex modal systems, thereby allowing designers and researchers to identify and eliminate mode errors before such systems are introduced to the marketplace.

ACKNOWLEDGMENTS

We thank Michael Heymann for his efforts in developing our understanding of non-deterministic systems in the context of human interaction with simple and complex devices. Part of this work was conducted under Grant NCC2-327 from NASA Ames Research Center, CA, to the San Jose State University.

REFERENCES

Aviation Week and Space Technology. (1995). Automated cockpits: who's in charge? Part 1 and 2. 142(5 and 6).

Billings, C. E. (1996). *Human-Centered Aviation Automation: Principles and Guidelines* (NASA Technical Memorandum 110381). Moffett Field, CA: NASA Ames Research Center.

Callantine, T. J., and Mitchell, C. M. (1994). A methodology for understanding how operators select and use modes of automation to control complex dynamic systems. *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1751-1756). San Antonio, Texas.

Degani, A., and Kirlik, A. (1995). Modes in human-automation interaction: Initial observations about a modeling approach. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (pp. 3443-3450). Vancouver, British Columbia, Canada.

Degani, A., Mitchell, C. M., and Chappell, A. R. (1995). Task models to guide analysis: Use of the operator function model to represent mode transitions. In R. S. Jensen (Ed.), *Proceedings of the Eighth International Aviation Psychology Symposium* (pp. 210-215). Columbus, OH: The Ohio State University.

Degani, A., Shafto, M., and Kirlik, A. (1995). Mode usage in automated cockpits: Some initial observations. In T. B. Sheridan (Ed.), *Proceedings of the International Federation of Automatic Control; Man-Machine Systems (IFAC-MMS) Conference*. Boston, MA: IFAC.

Degani, A., Shafto, G., and Kirlik, A. (1996). Modes in automated cockpits: Problems, data analysis, and a modeling framework. *Proceedings of the 36th Israel Annual Conference on Aerospace Sciences*. Haifa: Technion-IIT.

Eldredge, D., Dodd, R. D., and Mangold, S. J. (1991). *A review and discussion of flight management system incidents reported to the aviation safety reporting system*. Cambridge, MA: Volpe National Transportation Systems Center.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming, 8*, 231-274.

Harel, D. (1988). On visual formalisms. *Communications of the ACM, 31(5)*, 514-530.

James, M., McClumpha, A., Green, R., Wilson, P., and Belyavin, A. (1991). Pilot attitudes to cockpit automation. In R. S. Jensen (Ed.), *Proceedings of the Sixth International Aviation Psychology Symposium* (pp. 192-197). Columbus, OH.

Jones, P. M., Chu, R. W., and Mitchell, C. M. (1995). A methodology for human-machine systems research: Knowledge engineering, modeling, and simulation. *IEEE Transactions on System, Man, and Cybernetics, SMC-25(7)*, 1025-1038.

Minsky, M. L. (1967). *Computation: Finite and infinite machines.* Englewood Cliffs, NJ: Prentice-Hall.

Mitchell, C. M., and Miller, R. A. (1986). A discrete model of operator function: A methodology for information display design. I*EEE Transactions on Systems, Man, and Cybernetics, SMC-16(3)*, 343-357.

Monk, A. (1986). Mode errors: a user-centered analysis and some preventative measures using keying-contingent sound. *International Journal of Man-Machine Studies, 24*, 313-327.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 1(88)*, 1-15.

Sarter, N. B., and Woods, D. D. (1992). Pilot interaction with cockpit automation: Operational experience with the flight management system. *International Journal of Aviation Psychology, 2(4)*, 303-321.

Sarter, N. B., and Woods, D. D. (1994). Pilot interaction with cockpit automation II: An experimental study of pilot's mental model and awareness of the flight management and guidance system. *International Journal of Aviation Psychology, 4(1)*, 1-28.

Sarter, N. B., and Woods, D. D. (1995a). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors, 37(1)*, 5-20.

Sarter, N. B., and Woods, D. D. (1995b). *"Strong, silent, and 'out-of-the-loop'": Properties of advanced (cockpit) automation and their impact on human-automation interaction* (CSEL Report 95-TR-01). Cognitive Systems Engineering Laboratory, The Ohio State University, Columbus, Ohio.

Smith, S. L., and Gerhart, S. L. (1988). Statemate and cruise control: A case study. *Proceedings of the COMPAC '88, 12th IEEE International Computer Software and Application Conference* (pp. 49-56). New York: IEEE Press.

Vakil, S., Hansman, R. J., Midkiff, A. H., and Vaneck, T. (1995). Mode awareness in advanced autoflight systems. In T. B. Sheridan (Ed.), *Proceedings of the International Federation of Automatic Control; Man-Machine Systems (IFAC-MMS) Conference*. Boston, MA: IFAC.

Wiener, E. L. (1989). *The human factors of advanced technology ("glass cockpit") transport aircraft* (NASA Contractor Report 177528). Moffett Field, CA: NASA Ames Research Center.