ON THE TYPES OF MODES IN HUMAN-MACHINE INTERACTIONS

Ataf Degani
San Jose State University, CA, and NASA Ames Research Center, CA.

INTRODUCTION

'Mode': (1) manner of acting or doing; method, way. (2) the natural disposition as the manner of existence or action of anything; form. [Stevie's (1955) measure of manner. (Webster's Dictionary, 1994)]

The topic of modes has been widely acknowledged in the aviation psychology literature both as an important aspect of human-computer interaction and as a source of potential problems (Aviation Week and Space Technology, 1995; Funk, 1987; and Niemczyk, 1997). Several researchers have addressed the topic eloquently and provided, through a series of surveys, field studies, and experiments, much insight about the use of modes in 'glass cockpit' aircraft (Canner, 1994; Elderedge, Dodd, and Mangold, 1992; Sarter and Woods, 1994, 1995; Wiener, 1989). These studies and several highly publicized incidents and accidents have led to a wider recognition of the problem as an impediment to efficient and safe human-computer interaction (Abbott, Slote, and Stimson, 1996).

The topic of modes, however, is not only associated with cockpit automation. Many devices, machines, and computer interfaces exhibit modes. Human interaction with modes of word processors, VCRs, and military fire control systems have been addressed (Mark and Greer, 1992; Miller, 1979; Sellen, Kuntzhabach, and Burton, 1992). Although well studied and widely mentioned in the literature, a clear working definition of the term mode is lacking. Several authors define it very narrowly, some employ informal definition, while others defer definition and taxonomy for future research (Johansson, 1990, p. 424). The consequence, of course, is that different meanings are attached to the term depending on the authors' experience, the technology involved, and the domain's jargon.

Based on our research in modeling human interaction with modes, we employed a general working definition and developed a classification scheme of modes (Degani, 1996). This step was essential for the purpose of formally describing mode-based machines and identifying the features of the machine that may lead to mode ambiguity and subsequent classification. In this paper, we describe this classification and illustrate each type of mode using a modeling language called Structurals. Finally, we show how the classification and resulting modeling structures can be used for determining a portion of the speed and vertical modes of the automatic flight control system of a Boeing 757 aircraft.

TYPES OF MODES

In reviewing the literature on modes in human computer interaction (HCI) and human-machine system, we propose a classification of three types of modes: (1) Interface modes that specify the behavior of the interface; (2) Functional modes that specify the behavior of the various functions of a machine; and (3) Supervisory modes that specify the level of user and machine involvement in supervising the process.

Interface Modes

Tesler (1981) defines a mode in the context of an interactive system as a state of the user interface that lasts for a period of time, is not associated with any particular (display) object, and has no rule other than to place an interpretation on operator input. In the same spirit, Poller and Garter (1984) state that modes refer to the application's interpretation of the user's input—there is a mode change each time the interpretation of the same input changes. Neuringer (1978) notes that computing systems and their interfaces have many modes because different routines can be in control of the application—the mode specifies which routine is currently in control of the interface. For example, when a text editor is in "insert" mode, any text entry is inserted after the location of the cursor and existing text pushed forward; in "replace" mode, the new text replaces (and therefore deletes) the existing text. Such routines embody mode ambiguity. Monk (1986) argues that mode ambiguity exists when some action on the part of the user can have different effects on the computer depending on its current state (1986, p. 314). Mode ambiguity is the precursor for what Norman (1981) described as "mode error."

Figure 1 is an example of an interface mode that can be found in many computer applications and consumer electronic devices. The representation, or modeling, language we use here is Statecharts—a modern extension to the finite state machine (Harel, 1988). Three concurrent processes are depicted in this modeling structure of interface modes: button behavior, button indicator, and the display associated with this button. In this example we see that there is no indicator on the button itself (blank). In terms of button behavior, the button is either in its 'normal' state or 'pushed in'. Once the user pushed the button (to change a mode), the button releases (spring loaded) to the normal position. The display indicates the active mode—"On" or "Off". Initially, the display is "Off", but when the user pushes the button, the active mode changes to "On." Another such event (i.e., pushing the button) and it will go to "Off" again—the behavior of the display is circular. However, there could be yet another event that can...
cause a mode change. If event 'A' is sensed, an
appropriate transition to "On" takes place. Event 'A'
occurs somewhere also in our system, yet it impacts
the interface mode.

For example, consider a cordless phone that has
two primary states: "On" (line is open) and "Off".
The user controls this by a single (spring loaded)
button. When the phone rings, the user pushes the
button to listen and then hangs up by pressing the
same button again. Several modes of cordless phone
have an interesting feature: If the handset is on the
cradle, an incoming call arrives, and the user lifts up
the handset—the phone transitions from "Off" to
"On" automatically. Many users, while
knowledgeable about this feature, tend to push the
button upon hearing the ring—with the immediate
consequences of turning the phone "On" (and thereby
hanging up on the caller).

Functional

Mode discussion in the HCI literature focuses on
input/output relationships between the user and the
computer, specifically on data entry and the format of
the interface. When we survey the use of modes in
devices and machines, an additional type of mode
events. This mode refers to the active function of
the device. For example, many consumer electronic
devices have a variety of embedded functions: An
audio system may have a compact disk (CD) player,
cassette tape player, and radio. Selection of the
appropriate function is controlled via a mode switch,
which determines the functions, or modes, that are
currently active. Cook, Potter, Woods, and McDonald
(1991) evaluated the human engineering aspect of a
heated humidification device and identified several
mode related problems. For the purpose of their
study, they define a mode as "a device state which
controls or displays functions in a distinct way or has
distinct meaning" (p. 221). With this definition they
move one step beyond the HCI definition, which
focuses mostly on the distinct mode of the interface,
to dealing with devices that employ distinct
functions.

In dynamic control systems, mode behavior is
always a consequence of the process, and its
associated reference-value such as speed, temperature, or
time. Figure 2 is an example of a functional mode. Two
concurrent processes are depicted as this general
group of functional modes: modes and reference-
values. In the mode slide, we see that mode 1 is the
default mode (that is active anytime the machine is
started). Modes 2, 3, and 4 define the four different
types of behavior that this machine exhibits.
Transitions among modes 1, 2, and 3 are triggered
manually by the user (pushing a button) and are
labeled cm1, cm2, and cm3. An automatic transition
(e.g., triggered by an unusual event) can also take place.
Such transitions are depicted as broken lines—cmn.

The second process is the reference-value. Here
we have two states: manual and automatic. Initially,
the reference-value is manual. That is, the user
manually inserts the value (e.g., time) into the
interface. Once transition r, takes place, the source
of the reference-value is automatic—the machine uses
some stored value (or an algorithm) to provide this
value. There are many examples for this (mode and
reference-value) relationship in dynamic systems.
Consider for example the household microwave. The
microwave may have several cooking modes (defrost,
cook I, cook II). Always associated with such mode is
time (a reference-value). The source of the time
reference-value is usually manual—the user enters
this value to the interface. Nevertheless, when we
select popcorn, the time (reference-value) is set
automatically. The two elements—modes and its
reference-value—define the unique mode behavior of
the machine.

In complex and dynamic systems, both modes
and reference values can be changed automatically,
manually, or by default. Such interdependency
between modes and reference values appears to be a
consistent source of mode ambiguity.

For example, Guba (1994) reported an incident
with an automatic blood pressure machine device.
During an operation, the anesthesiologist changed the
reference-value (interval between measurements) not
realizing that this drove the machine into an undesired
"idle" mode (without any clear announcement of this
mode transition).

Supervisory

To our departure from the opening of modes in
HCI, we now add another type of mode—supervisory
—sometimes also referred to as participatory modes
(Wickens and Kessel, 1979). Complex machines that
operate in an uncertain environment usually allow the
user to specify the level of human-machine
involvement in controlling the process (McDaniel,
1988). For example, on June 4, 1996, a U.S Navy
aircraft was shot down during exercises while having
a target for ship-to-air gunnery practice (Proctor,
1996). An anti-aircraft fire control system onboard
a naval vessel (the Japanese destroyer Yugary) transitioned
from manual (optically guided) tracking mode to automatic (radar guided). In the process, the
operator locked sighted with the arrow in the radar display.

Figure 3 is an example of a supervisory mode that
is based on many control systems, such as cruise
control of an automobile, robots on assembly lines, and aircraft
flight control systems. This hypothetical machine has
four levels of automation. The first level is manual.
The user operates the components manually and the
Automatic control system is "Off." Second level is when the user still operates manually, yet the automatic system is working in the background (and possibly giving the user some guidance). Once the control system is engaged, a variety of semi-automatic modes are available. This is the third level. The last level is the fully automatic mode.

For example, consider automatic cruise control of a modern car. At the first level, the driver controls the speed manually. He or she may then arm the cruise control ("On"), yet continue driving manually. Once the "set" button is pushed, the cruise control automatically maintains the current speed of the car. This is the highest level of automation in cruise control. At some point, the driver may want to temporarily accelerate (e.g., pass a car ahead) and then return to the previous cruising speed. This is a manual override mode. However, once the driver releases the gas pedal the car returns to the previously set speed. All these levels define different combinations of human and machine involvement in controlling the process.

Summary

The literature review alludes to three different types of modes that are employed in computers, devices, and supervisory control systems: interface modes are used for data entry and changes in display format; supervisory/interaction modes (manual, semi-automated, and fully automated) are used for data entry and changes in display format; supervision modes (manual, semi-automated, and fully automated) are used for data entry and changes in display format.

The three sub-types are essentially similar; they all define a manner in which a certain component of the machine behaves. The component may be the interface only, the machine, and the level of supervision exhibited by the human supervisor. This commonality brings us to a definition of the term "mode." As a general working definition, we define mode as the manner of behavior of a machine (Ashby, 1956/1968, chap. 4). While this general definition is quite broad, the three mode categories proposed here constrain the definition and provide us a foundation for modeling and analysis.

Auto Flight Example

The Automatic Flight Control System (AFCS) of a Boeing 787 has eight functional modes to control the vertical aspect of the flight path. These modes are discussed here: "Altitude Capture," "Altitude Hold," and "Vertical Navigation." With respect to the speed reference value, the "Altitude Capture" and "Altitude Hold" modes obtain this parameter from the mode control panel, which displays the current speed at the beginning of the altitude-capture maneuver. By default, the "Vertical Navigation" speed reference value is contained from the flight management computer, which computes the most economical speed for the particular flight regime. Yet another option, called "Speed Intervene," allows the pilot to override the flight management computer speed and manually enter a different speed. This is achieved by pressing the "speed" knob and then dialing-in the desired speed to the mode control panel. Figure 4 is a simplified model of this machine. It contains two modules: the interface element (a portion of the mode control panel) and the control mechanism itself.

Interface

In the upper module in Figure 4 we describe the speed knob and speed window located on the mode control panel. In describing the speed knob we employ the modeling structure that was described earlier. The speed knob has two states: "normal" and "pushed-in." The initial state of the speed knob is "normal," but when momentarily pushed, it engages or disengages the "speed intervene" sub-mode of the "vertical navigation" mode. The speed window display can be either open or closed. When in "vertical navigation" mode this is achieved by pressing the speed button (c). The speed knob behavior is circular, but similar to our example of the cordless phone, an external event (not in VNAV) can also prompt a transition.

Control Mechanism

The mode structure of the AFCS contains both functional and supervisory modes. We shall first define functional modes and their reference values and then the supervisory modes.

In the control mechanism we describe the "vertical navigation" (VNAV) modes: "vertical speed," "altitude hold," and "altitude capture" modes (Figure 4). All are functional modes in the vertical aspect of the flight. As mentioned earlier in our discussion of dynamic control systems, a (functional) mode is always associated with a reference value. The reference-value element of the vertical modes contains two states: "flight management computer" and "mode control panel." Each describes the source for the speed reference value. Initially, the source of the speed parameter is from the mode control panel. Engagement of "vertical navigation" via the mode control panel will cause a transition to the flight management computer as the source of speed values (transition νv). Engagement of any one of the semi-automated modes (each of which "altitude hold") causes a transition to the mode control panel as a source for the speed-reference value (νv). As mentioned above, the crew has an option for manually adjusting the speed. A sub-mode of "vertical navigation"—namely "speed intervene"—allows the pilot to enter speed values via
the mode control panel and hence override the speeds computed by the flight management computer. This is accomplished by pressing the speed knobs located on the mode control panel (c). The relationship between the functional modes and their reference-value in this machine is characteristic of many dynamic control systems. Similar to the automatic blood pressure machine example, the mode behavior is complex, and may lead to confusion (Degani and Kirlin, 1995).

The supervisory mode levels in the AFCS are organized hierarchically. Two main levels are described in Figure 4: fully automatic and semi-automatic. The highest level of automation in this partial description is the "Vertical Navigation" mode. This mode is depicted as a state at the top of the mode pyramid in Figure 4. "Vertical Navigation" mode is both functional and supervisory. It is functional in that it has a unique mode behavior in terms of control algorithms and reference values. It is also a distinct supervisory mode (fully automatic) as the crew can pre-program it in advance and let the computer completely manage the navigation of the flight. In contrast, "Vertical Speed" and "Altitude Hold" are semi-automatic modes—the crew must constantly manipulate them in order to navigate the aircraft.

Supervisory Mode Structure

Other modes in the AFCS can only be engaged automatically—no direct manual engagement is possible. One such example is the "Altitude Capture" mode. This mode engages automatically only when the aircraft is climbing or descending to an altitude. When the aircraft is close to the desired altitude, an automatic transition to the vertical mode (e.g., "Vertical Navigation") to "Altitude Capture" takes place (conc.). This transition ("Vertical Navigation" -> "Altitude Capture") will trigger an automatic transition in the reference value element. The source of the speed reference-value will change from 'Flight management computer' to 'mode control panel'.

In summary, the relationship between the interface modes, functional modes and reference-value, and supervisory modes in the AFCS of the B-757 are far from trivial. Transitions between mode and the resulting effect on speed reference-value are dependent on a variety of events, some of which are manual (engage/disengage "Vertical Navigation" via the mode control panel), some are dependent on the previous mode (engage/disengage "Speed Intervene"), and others are automatic (transition from "Vertical Navigation" to "Altitude Capture" and then to "Altitude Hold").

CONCLUSIONS

In this paper we have briefly discussed human interaction with modes from a perspective of different domains and machines. We proposed a general definition of the term "mode" and then a more constrained sub-classification. We illustrate each type of mode by describing its unique modeling structure and note some of its characteristics. We then proceeded to formally describe a project of the interaction between the pilot and the automatic flight control system of a Boeing B-757. We note that all three types of modes exist here. Interface mode specifying the behavior of the human on the mode control panel, functional modes and their associated speed reference-values; and supervisory modes with their different levels of human-machine involvement.

The classification of modes proposed here is not mutually exclusive. We saw one case in which one mode ("Vertical Navigation") was both functional and supervisory. Some modes are induced by an intersection of two mode types and that is an important attribute of the human-machine system. Furthermore, the classification of modes provided us with modeling structures (we call them templates) that can be consistently employed in describing mode-based control systems. In our research we have used these templates to model a variety of human-machine systems from simple consumer electronic products, to more complex and automated devices, and all the way to avionics systems (Degani, 1996). It allowed us to describe and then identify the features of the machine that lead to mode ambiguity and error.

REFERENCES


962


Miller, R. A. (1979). Identification of finite state models of human operations (Dept of Industrial and systems Engineering project # 784556 AFOSR, #75-3155). Columbus, OH: The Ohio State University.


![Figure 1: Interface Mode](image-url)