Published in the Proceedings of the 21st Digital Avionics Systems Conference (DASC'02), Irvine, California, Oct. 27-31, 2002. *Selected as best paper of the Software Engineering for Human Systems session.*

A METHODOLOGY FOR IMPROVING MODE AWARENESS IN FLIGHT GUIDANCE DESIGN

Steven P. Miller, Sarah Barber, Timothy M. Carlson, David L. Lempia, Alan C. Tribble

Rockwell Collins Inc, Cedar Rapids, Iowa

Introduction

Digital avionics systems such as Ground-Proximity Warning Systems (GPWS), Traffic Collision Alert and Avoidance Systems (TCAS), and Weather Radar (WRX) have made possible many of improvements in air safety seen over the last few decades [1]. Improving aviation safety while air traffic increases by an order of magnitude over the next two decades [2] will require still more sophisticated systems.

However, the levels of integration, automation, and complexity of today's systems also place greater cognitive demands on the flight crew [3]. Pilots must master several complex, dynamically interacting systems, often operating at different levels of automation, that have evolved over a number of years. To provide maximum flexibility and to accommodate the variety of situations that occur during flight, these systems often have many different modes of operation, each with different responses to crew actions and other systems.

Mode confusion occurs when the flight crew believes the automation is in a mode different than the one it is actually in and consequently make inappropriate requests or responses to the automation. Mode confusion can also occur when the flight crew does not fully understand the behavior of the automation in certain modes or how different modes interact, i.e., when the crew have a poor "mental model" of the automation [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15].

From early indications in the 1980's [16], [17], there has been growing evidence that mode confusion is an important safety concern. Several aircraft accidents and incidents involving mode confusion are listed in [18]. A study conducted by the Massachusetts Institute of Technology found 184 incidents attributed to mode awareness problems in NASA's Aviation Safety Reporting System (ASRS) [18]. An FAA workshop on Autoflight Mode Awareness identified "autoflight mode confusion as a significant safety concern" [19]. The Loss of Control Joint Safety Analysis Team (JSAT) chartered by the Commercial Aviation Safety Team (CAST) [20] identified improved training of automated flight systems as one of their top recommendations.

Many researchers fault automation that is overly complex, inconsistent, and does not provide the pilots with the information they need in a form they can easily find and digest. According to Charles Billings, "*the central technical (and even social and legal) problems for the human operators who work with today's automation are the complexity and opacity of these tools*" [21]. Increasingly, researchers and pilots are calling for "human centered automation" that provides transparent and consistent behavior that supports the operators in their monitoring and control tasks [4], [13], [21].

To do this, pilots and human factors engineers must be able to understand and critique a system's behavior. At the same time, the system and software engineers need to better understand how their design decisions may contribute to mode confusion. Rockwell Collins and the NASA Langley Research Center are jointly sponsoring a project to identify methods and tools that can help address both of these needs. This project makes use of both traditional approaches such as checklists and design reviews, and advanced techniques such as formal modeling, model-checking, and theorem proving [23], [24], [25], [26], [27], [27].

This work was supported in part by the NASA Aviation Safety program and the Langley Research Center under Contract NCC-01-001.

This paper describes the application of an informal approach to identify and reduce potential sources of mode confusion early in the development cycle when changes can be made at reasonable cost. A checklist of potential sources of mode confusion was developed through a literature review, then applied to a representative specification of a Flight Guidance System documented in the RSML^{-e} notation [28]. The potential sources of mode confusion identified through use of the checklist were then reviewed with pilots, engineers, and other domain experts. While most of the issues identified were determined to be benign, or even natural consequences of conforming to existing aviation conventions, several concrete recommendations were made to improve mode awareness. Surprisingly, most of these required only minor changes, yet provided significant improvement in mode awareness. The full report [29] is available from the NASA Langley Research Center.

The next sections provides a brief overview of a Flight Guidance System and the RSML^{-e} specification language. The main body of the paper discusses the mode confusion taxonomy and its application to the specification of the FGS, followed by two examples of recommendations that were generated by the review process. The last section presents conclusions and directions for future work.

Overview of a Flight Guidance System

A Flight Guidance System (FGS) is a component of the overall Flight Control System (FCS). It compares the measured state of an aircraft (position, speed, and attitude) to the desired state and generates pitch and roll guidance commands to minimize the difference between the measured and desired state. The internal logic of an FGS can be broken down into the mode logic and the flight control laws. The flight control laws accept information about the aircraft's current and desired state and compute the pitch and roll guidance commands. The mode logic determines which lateral and vertical modes are active and armed at any given time. These in turn determine which flight control laws are active and armed.

The flight crew interacts with the FGS primarily through the Flight Control Panel (FCP) (see Figure 1), which provides controls for selecting the various modes of operation and setting reference values such as desired airspeed or altitude. The FCP also supplies feedback to the crew, indicating selected modes by lighting lamps on either side of a selected mode's button.

The current and armed modes are also annunciated on the Primary Flight Displays (PFD) along with a graphical depiction of the flight guidance commands generated by the FGS (see Figure 2). The PFD displays essential information about the aircraft, such as airspeed, vertical speed, attitude, the horizon, and heading. The active lateral and vertical modes are displayed (annunciated) at the top of the display. The annunciations in Figure 2 indicate that the current active lateral mode is Heading Select (HDG), the active vertical mode is Pitch (PTCH), and that Altitude Select (ALTS) mode is armed.

The large sphere in the center of the PFD is the sky/groundball. The horizontal line across its middle is the artificial horizon. The current pitch and roll of the aircraft is indicated by a white wedge ^ representing the aircraft in the middle of the sky/ground ball. Figure 2 depicts an aircraft with zero degrees of roll and pitched up approximately five degrees.

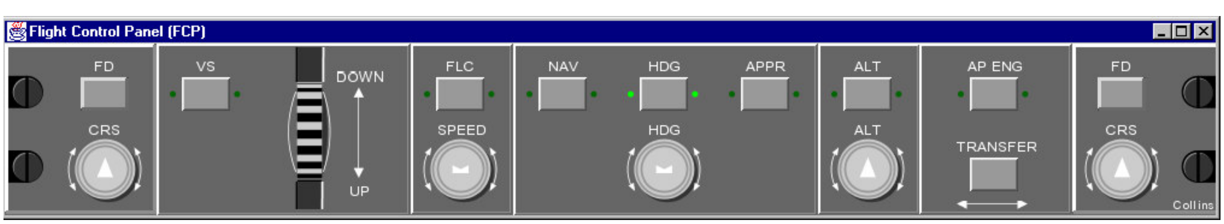

Figure 1 - Flight Control Panel

The graphical presentation of the pitch and roll *guidance* commands on the PFD are referred to as the Flight Director (FD)[1], and are shown as a magenta wedge ^ on the sky/ground ball. When the autopilot is not engaged, these are interpreted as guidance to the pilot. When the autopilot is engaged, these indicate the direction the aircraft is being steered by the autopilot. Figure 2 depicts an aircraft in which the autopilot is not engaged and the Flight Director is commanding the pilot to pitch up and roll to the right.

Figure 2 - Primary Flight Display

The RSML^{-e} Specification Language

RSML (Requirements State Machine Language) is a state-based specification language developed by Nancy Leveson's group at the University of California at Irvine as a language for specifying the behavior of process control systems [30]. One of the main design goals of RSML was readability and understandability by non-computer professionals such as end-users, engineers in the application domain, managers, and representatives from regulatory agencies. RSML was used to specify TCAS-II and this specification was ultimately adopted by the FAA as the official specification for TCAS-II. RSML was heavily influenced by Statecharts [31] and uses a similar notion of explicit event propagation. In the course of developing the TCAS-II specification and the subsequent independent verification and validation effort, it became clear that the most common source of errors was this dependence on explicit events [33]. To eliminate this problem, the Critical Systems group at the University of Minnesota developed RSML^{-e} (RSML without events) [28]. As its name implies, RSML^{-e} eliminates the use explicit events and is a synchronous language [32]. RSML^{-e} is similar to another derivative of RSML, SpecTRM-RL, developed by the Safeware Engineering Corporation, but has a slightly different syntax and semantics and differs in the underlying modeling philosophy.

The Mode Confusion Taxonomy

As a first step to understanding what causes mode confusion, a survey of the literature was conducted and distilled into a taxonomy of common causes of mode confusion in system designs [34]. While compiled from a number of sources, the taxonomy is heavily based on one described by Leveson, et al. in [7] and on the studies of Sarter and Woods in Woods [10], [11], [12], [13], [14], [15]. The main categories of the taxonomy are listed in Table 1 in the next section. To make the taxonomy more directly useful to developers, it was extended with a checklist of specific items to look for during review of a requirements or design specification. It quickly became apparent that the checklist lost much of its value when separated from the literature survey. Examples and rationale were necessary to illustrate why the checklist items might indicate a potential source of mode confusion. As a result, a format was developed that combined the taxonomy, illustrative examples, and references to the literature along with the checklist. This document is available from the NASA Langley Research Center [34].

To assess the usefulness of the taxonomy, it was applied to an example specification of the

[1] The term Flight Director is also commonly used to refer to the logic that computes the pitch and roll guidance commands.

mode logic of a Flight Guidance System written in the RSML^{-e} requirements specification language [35]. This took only a few hours for one person to complete, yet resulted in a list of over seventy issues to look at more closely. The format of the RSML^{-e} specification made identification of these issues straightforward, often almost mechanical.

However, it also became apparent that violation of one of the checklist items did not always indicate a source of mode confusion. For example, indirect mode transitions (transitions between system modes that are not triggered by an immediate action of the operators) are a common source of mode confusion, but as pointed out in [7], it would be impossible to build a modern avionics systems without them. However, it is important to determine whether the operators are able to predict when indirect mode transitions will occur and whether they are appropriately notified when an indirect mode transition occurs. Identifying indirect mode transitions from a formal specification is easy. Determining whether sufficient feedback is provided to the operators can require considerable domain expertise.

As another example, side effects such as changing a mode or a data value in one subsystem when a change is made to another subsystem are another frequent source of mode confusion. However, Sarter and Woods noted cases in which the pilots were confused when a side effect did not occur, specifically when the system failed to initiate a mode transition after a target value was changed [13] or failed to propagate a changed value from one part of the system to another [10]. The key here appears to be to determine whether the operator expects the side effect to occur as a result of his or her actions. Again, identifying side effects is relatively easy. Determining whether they are actually a potential source of mode confusion requires careful review.

To determine which of the issues raised by the checklist were significant, a small group of engineers and pilots was formed to review and discuss the most important issues. The results of these reviews were captured in an Issue Based Information System (IBIS) [36]. This IBIS also captured specific recommendations made by the group.

Taxonomy Category	Issues Raised by	
	Checklist	**Discuss**
1.Transitions Between Normal and Off-normal Modes	6	-
2. Interface Interpretation Errors		
2.1 Input Interface Interpretation Errors	2	-
2.2 Output Interface Interpretation Errors	-	1
3. Inconsistent Behavior	3	-
4. Indirect Mode Changes	15	2
5. Operator Authority Limits	-	-
6. Ignored Operator Commands	12	1
7. Hidden Modes	5	1
8. Side Effects	32	-
9. Lack of Appropriate Feedback		
9.1 Lack of Feedback on Current Modes	3	2
9.2 Lack of Feedback on Pending Mode Changes	2	1
9.3 Lack of Feedback with Multiple Operators	-	1
9.4 Lack of Feedback from Independent, Redundant Sources	-	-
9.5 Loss of Feedback from Implicit Sources	-	-

Table 1 – Potential Sources of Mode Confusion Identified by Taxonomy Category

Published in the Proceedings of the 21st Digital Avionics Systems Conference (DASC'02), Irvine, California, Oct. 27-31, 2002. *Selected as best paper of the Software Engineering for Human Systems session*.

While many of the issues were deemed to not be significant sources of mode confusion, several recommendations were made that the group felt would reduce the potential for mode confusion. Surprisingly, the recommendations were often relatively minor changes.

While many of the checklist issues were not significant sources of mode confusion, the list of issues provided an essential focus for the reviews. When meetings were held without specific checklist issues to discuss, relatively little was accomplished. However, it was also important to allow the reviewers to deviate from the existing list of issues. Often, during the discussion of an issue, the reviewers would identify other potential sources of mode confusion that had not been identified using the checklist. Since these were based on their expertise, they were often more significant than the original issues. Thus, the list of issues worked both as a meeting agenda to keep the group focused and as a catalyst for identifying new issues. Table 1 summarizes by taxonomy category the potential sources of mode confusion identified in this exercise, including those first raised through use of the checklist and those raised during review by the domain experts.

Table 1 clearly indicates that some taxonomy categories were more useful when reviewing the specification than others. For example, 32 instances of side effects were discovered while no instances of output interface errors, lack of feedback from independent redundant sources, or loss of feedback from implicit sources were identified. The reasons for this are discussed in the following sections.

Side Effects

The largest number of potential sources of mode confusion were raised by examining the specification for side effects. In large part, this was because the tabular format of the RSML^{-e} specification made the identification of side effects straightforward. For example, to identify the side effects associated with the Flight Director (FD) guidance cues, it was only necessary to review the tables that define when the cues are turned on and turned off. The RSML^{-e} definition of when the FD cues are to be turned on is shown in Table 2.

Table 2, referred to as an And-Or table, lists in the left-hand column the events or conditions that are inputs to the macro. (A *condition* is a Boolean valued function that is true or false at a point in time, i.e., a predicate over a single state. An *event* is a condition that was false in the previous step and true in the current step, i.e., a predicate over two states.) Each of the remaining columns to the right specify which of those events or conditions need to be true or false, indicated by a T or an F in the appropriate cell. A dot in a cell indicates a don't care condition. Each column defines the conjunction (and) of the events or conditions that need to met for the macro to be true, and the array of columns represent the disjunction (or) of these events/conditions.

					OR			
	When_FD_Switch_Pressed()	T	•	•	•	•	•	•
	When (AP = Engaged)	•	T	•	•	•	•	•
	When (Overspeed)	•	•	T	•	•	•	•
A	When_GA_Switch_Pressed()	•	•	•	T	•	•	•
N	When_Lateral_Mode_Manually_Selected()	•	•	•	•	T	•	•
D	When_Vertical_Mode_Manually_Selected()	•	•	•	•	•	T	•
	When_Pilot_Flying_Transfer()	•	•	•	•	•	•	T
	Pilot_Flying = THIS_SIDE	•	•	•	•	•	•	T
	Were_Modes_On()	•	•	•	•	•	•	T

Table 2 - Definition of When the FD Cues Are Turned On

For example, the Flight Director guidance cues are turned on when the flight director switch is pressed, or when the Autpilot (AP) is engaged, or when an overspeed condition exists, etc. All the rows in Table 2 represent events except for the last two, *Pilot Flying = THIS_SIDE* and *Were_Modes_ On()*, which are conditions. The last column states that the Flight Director guidance cues should be turned on when there is a transfer of control to this side of the aircraft while the mode annunciations are turned on.

Identifying side effects from Table 2 is straightforward once a small bit of subjective judgment is provided. Clearly, turning the Flight Director guidance cues on when the FD switch is pressed is not a side effect – the pilot is explicitly requesting the cues to be turned. On the other hand, turning the FD cues on when the autopilot is engaged, however reasonable, is technically a side effect of engaging the autopilot. In fact, turning the FD cues on is a side effect of all the other events listed in Table 2. Examination of Table 2 thus leads to the following list of side effects

- FD turns on when the Autopilot is engaged.
- FD turns on when an overspeed condition occurs.
- FD turns on when the Go Around (GA) switch is pressed.
- FD turns on when a lateral mode is manually selected.
- FD turns on when a vertical mode is manually selected.
- FD turns on when there is a pilot flying transfer while the mode annunciations are on and the transfer is to that side.

Whether these side effects are actually sources of mode confusion can only be determined through careful review with domain experts. The intent is clearly to display the FD cues when the flight crew select a new mode of operation, when an overspeed condition occurs, when the autopilot is engaged, or when there is a transfer of control to the other pilot. It can be argued whether displaying the cues when selecting a new mode or engaging the autopilot improves mode awareness or not. Displaying the cues when an overspeed condition occurs seems reasonable, but might also be subject to debate as the crew receives several other alerts during an overspeed condition. In similar fashion, it can also be argued whether display of the cues should be transferred to the display of the new pilot flying side. These are all judgment calls and the rightful province of pilots and other domain experts.

One of the reasons that a large number of side effects were discovered during application of the check lists is that the RSML^{-e} format lends itself well to the identification of side effects. Also, the mechanical way in which this can be done suggests that this process can be automated. If information was provided by the specifier as to which were the primary triggering events (e.g., pressing the FD button), the remaining events could be identified by a tool as potential side effects.

Ignored Operator Commands

The next largest number of issues were raised by examining the specification for ignored operator commands. These were found by scanning the RSML^{-e} specification for operator initiated events and looking for guard conditions that would inhibit the handling of the event. For example, Table 3 defines when Vertical Speed (VS) mode is to become active.

A	When_VS_Switch_Pressed()	T
N	Overspeed	F
D	Was_VAPPR_Active()	F

Table 3 – Vertical Speed Selection

In this case, Vertical Speed mode is to become active when the VS switch is pressed by the pilot, providing that an overspeed condition does not exist and Vertical Approach mode was not active at the start of the step. This leads to two instances of an ignored operator command:

- Vertical Speed mode cannot be selected by pressing the VS switch during an overspeed condition.
- Vertical Speed mode cannot be selected by pressing the VS switch while Vertical Approach mode is active.

Even if this is the appropriate system behavior, the question immediately arises as to whether the pilot is provided with sufficient information to understand why his or her request is being rejected. In both cases, several other visual and aural cues are provided to the pilot that an overspeed condition exists or that Vertical Approach mode is active. Would displaying additional information about why a request for Vertical Speed is being ignored be useful, or would it be information overload?

As with the identification of side effects, ignored operator commands can be easily identified once some subjective judgment is applied. In Table 3, the key was recognizing that pressing the VS switch is an operator initiated event. The guard conditions then indicate that this request to be ignored. If the specifier were to annotate the events to distinguish which are initiated by the operator, it should be possible for a tool to identify all cases in which these requests would be ignored.

Other Taxonomy Categories

While space does not permit a discussion of each of the taxonomy categories, a more detailed discussion can be found in [29]. However, it is worth briefly discussing a few of the categories for which no issues were raised by the inspection.

No issues were identified under output interface interpretation issues largely because the specification defined few outputs other than the mode annunciations themselves. Issues related to the mode annunciations were recorded under the categories of lack of feedback on current and pending modes. This was probably an anomaly of the particular specification rather than an indication of a problem with the category.

No issues were recorded under the categories lack of feedback from multiple operators (although one issue was raised during group discussion), lack of feedback from multiple, redundant sources, and loss of feedback from implicit sources. There does not appear to be anything inherent in the $RSML^{-e}$ language that prompts their identification analogous to that found for other categories such as side effects and ignored operator commands. This may also indicate that other sources than a specification of the planned system behavior need to be examined to identify such sources of mode confusion. More thought needs to be given to how these issues could be identified during the inspection process.

Examples of Recommendations

In this section, two examples of recommendations arising out of this process are given. The purpose here is to illustrate the sort of changes that arise from this process and to show how easily these changes could be made if they were identified early in the life cycle. These examples are of particular interest since they occur widely across the industry.

Mode Annunciations

In most flight guidance systems, the active and pending modes are annunciated in text fields at the top of the primary flight display (Figure 3). The lateral and vertical capture annunciations indicate the current active lateral and vertical modes, while the lateral and vertical armed annunciations indicate the armed lateral and vertical modes (if any). Active modes are normally annunciated in one color (often green), while armed modes are annunciated in another color (often white). When an armed mode becomes active, the armed field is cleared and the text is moved to the capture field and changed from white to green, indicating that the mode has transitioned from the armed to active state. While the text, color, and placement of the annunciations vary across the industry, most commercial systems annunciate the modes in this fashion. Figure 3 illustrates the mode annunciations when Heading Select (HDG) is the active lateral mode, Vertical Speed (VS) is the active vertical mode, a lateral mode is armed to capture the VOR, and the vertical mode of Altitude Select (ALTS) is armed to capture the preselect altitude.

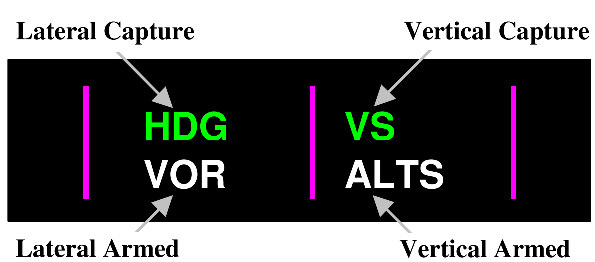

Figure 3 - Mode Annunciations

Published in the Proceedings of the 21st Digital Avionics Systems Conference (DASC'02), Irvine, California, Oct. 27-31, 2002. *Selected as best paper of the Software Engineering for Human Systems session*.

To convey additional information to the pilot, a common industry convention is to indicate some modes by annunciating their navigation source. For example, the annunciation of VOR is displayed in the lateral armed field when either the lateral navigation or lateral approach mode is armed with a VOR as the selected navigation source (see Figure 3).

This can be confusing if different modes use the same navigation source. In the example shown in Figure 3, it is not immediately obvious whether lateral approach or lateral navigation mode is armed for the VOR. While this indication is clearly provided on other displays (for example, the lamps on the Flight Control Panel shown on Figure 1), it would be even clearer to incorporate this information directly into the annunciation.

The recommendation that grew out of the review was to add a second field to each mode annunciation that indicates the navigation source or reference associated with the mode. For example, the annunciation of Figure 3 would be rendered as shown in Figure 4.

Figure 4 – Mode and Navigation Source

The text NAV is displayed as the armed lateral mode. Its navigation source, VOR, is indicated immediately following. The same text would be used to label both the mode annunciation (in this case NAV) and the FCP switch. In this way, the pilot can expect that if s/he presses the switch labeled X, the annunciation X will appear on the display. In the same way, to clear mode X, the pilot can always press the switch labeled X on the FCP.

This solution may not have been feasible on the smaller displays of a decade ago. However, as displays grow larger and cheaper, it seems reasonable to devote more space to this key piece of information. For modes that do not have an associated navigation source, such as HDG or VS, the second field can be used to display other information related to the mode, such as the current heading or the vertical speed reference. Of course, consideration should be given to whether this is useful to the pilot or merely clutters the display with information available elsewhere.

Capture Predicted Indication

A common form of mode confusion leading to "altitude bust" has been described in a number of papers [6], [9], [25], [26] [27]. In the typical scenario, the Flight Guidance System is armed to capture a target altitude such as the Preselect Altitude. As the aircraft nears the target altitude, it enters a capture mode that begins to level the aircraft off. The existence of this submode may or may not be clearly annunciated to the flight crew depending on the specific aircraft. While in the capture mode, the pilot changes the target altitude, perhaps as directed by Air Traffic Control, to climb or descend to a new altitude. Since attempting to capture the new target altitude while in the capture submode may result in excessive accelerations, most systems first place the aircraft into a new vertical mode such as Pitch Hold or Flight Level Change. However, if the system is placed in a mode (e.g., Pitch Hold) that flies the aircraft away from the new target altitude, the aircraft may fail to capture the target unless the pilot first changes the vertical direction of the aircraft. For example, consider the case in which the aircraft is ascending during capture of the target altitude and the pilot changes the target altitude to below the aircraft. This drops the system into Pitch mode such that the aircraft continues ascending. Unless the pilot notes that the new target altitude will not be captured without a change of vertical direction, this could lead to an "altitude bust" in which the aircraft departs from its assigned airspace. In [25], the pilots surprised by this behavior viewed it as a design flaw that should be corrected.

This behavior was also present in the example specification, though with some differences from that cited in [25]. While there are several possible solutions, the recommendation made by the group was similar to that suggested by Rushby in [26]. This consisted of adding a "capture predicted" indication near the Preselect Altitude on the Primary Flight Display. This would indicate if the aircraft was flying towards the target altitude so that capture is ultimately possible. For example, Figure 5 shows a display in which the Preselect Altitude of

2,200 feet is enclosed by a box indicating that it will be captured since Altitude Select mode is armed for the Preselect Altitude (ALTS), the aircraft is below the 2,200 feet, and the aircraft is flying towards the Preselect Altitude. If instead the aircraft was descending and flying away from the Preselect Altitude, the box would be turned off, indicating to the pilot that capture would not occur.

Figure 5 – Preselect Altitude Capture Predicted

Conclusions and Future Directions

This effort demonstrates that it is possible to identify potential sources of mode confusion early in the product life cycle, and to do so at relatively low cost. It reinforces the hypothesis of Sarter and Woods [10], [11], [12], [13], [14], [15]and Leveson [7] that certain patterns of functional behavior are indicative of potential sources of mode confusion, and that these patterns can be identified in requirements and design models. The taxonomy of mode confusion sources proposed by Leveson [7] was refined and extended with a checklist of items that could be used when reviewing requirements and design specifications. The use of formal models, rather than natural language specifications, made use of the checklist much simpler.

A shortcoming of this approach is that it frequently indicates potential sources of mode confusion when there are none. Separating the true sources of mode confusion from desirable system behavior requires that each issue raised through use of the checklist be reviewed by domain experts. While not optimal, this review process does lead to the discovery of other potential sources of mode confusion not identified with the checklist. Since these are based on the reviewer's expertise, they also tend to be actual sources of mode confusion. Use of the checklist serves both to guide the reviews and as a catalyst for discovery.

One area for future work would be to refine the checklist so as to not generate as many false indications. Preliminary efforts to do this also eliminated much of the benefits of the approach, suggesting that this may not be a trivial exercise. In addition, no potential sources of mode confusion were identified for some categories. This suggests that more thought needs to be given as to how such sources of mode confusion might be identified.

The straightforward way in which some potential sources of mode confusion were identified suggests that if the models were annotated with a small amount of additional information, it would be possible to detect potential sources of mode confusion with rather simple automated tools. An exciting area for further work is to use more powerful analysis tools, such as model checkers or theorem provers, to better identify potential sources of mode confusion.

References

[1] Earl F. Weener, Commercial Transport Safety, Airliner, Boeing Commercial Airplane Group, pg. 1-9, April-June 1993.

[2] U.S. Federal Aviation Authority, Office of System Safety, The Global Aviation Information Network (GAIN): Using Information Proactively to Improve Aviation Safety, February 2002,

http://www.asy.faa.gov/gain/What_Is_GAIN/whatisgain.htm.

[3] Anonymous, Flight Guidance System Approval, Joint Advisory Circular AC/ACJ 25.1329, Draft 14, July 3, 2001.

[4] J. M. Carroll and J. R. Olson, Mental Modes in Human-Computer Interactions, Handbook of Human-Computer Interaction, M. Helander (Ed.), Elsevier Science Publishers, pp. 45-65, 1988.

[5] Asaf Degani, Modeling Human-Machine Systems: On Modes, Error, and Patterns of Interaction, Ph.D. thesis, Georgia Institute of Technology, 1996.

[6] Asaf Degani and Michael Heymann, Pilot-Autopilot Interaction: A Formal Perspective, in Proceedings of the International Conference on Human-Computer Interaction in Aeronautics: HCI-Aero 2000, Toulouse, France, September, 2000.

[7] Nancy Leveson, et al, Analyzing Software Specifications for Mode Confusion Potential, in Proceedings of a Workshop on Human Error and System Development, C.W. Johnson, Editor, pg. 132-146, Glasgow, Scotland, March 1997.

[8] Nancy G. Leveson, Safeware: System Safety and Computers, Addison-Wesley Publishing Company: Reading, Massachusetts, 1995.

[9] Nancy G. Leveson, Designing Automation to Reduce Operator Errors, in Proceedings of the IEEE Systems, Man, and Cybernetics Conference, October 1997.

[10] Nadine D. Sarter and David D. Woods, Pilot Interaction with Cockpit Automation: Operational Experiences with the Flight Management System, The International Journal of Aviation Psychology, 2(4), pg. 303-31, 1992.

[11] Nadine B. Sarter and David D. Woods, Pilot Interaction with Cockpit Automation II: An Experimental Study of Pilots' Model and Awareness of the Flight Management System, The International Journal of Aviation Psychology, 4(1), pg. 1-28, 1994.

[12] Nadine D. Sarter and David D. Woods, Decomposing Automation: Autonomy, Authority, Observability and Perceived Animacy. First Automation Technology and Human Performance Conference, April 1994.

[13] Nadine D. Sarter and David D. Woods, "How in the World Did I Ever Get Into That Mode?", Mode Error and Awareness in Supervisory Control, Human Factors, 37(1), pg. 5-19, 1995.

[14] Nadine D. Sarter and David D. Woods, "Strong, Silent, and Out-of-the-Loop", CSEL Report 95-TR-01, Ohio State University, February 1995.

[15] Nadine D. Sarter and David D. Woods, and C. E. Billings, Automaton Surprises, in Handbook of Human Factors/Ergonomics, 2nd Edition, G. Salvendy (editor), Wiley: New York, 1997.

[16] E. L. Wiener, Human Factors of Advanced Technology ("Glass Cockpit") Transport Aircraft, NASA Contractor Report 177528, Moffett Field, CA: NASA-Ames Research Center, 1989.

[17] Dan Hughes, Glass Cockpit Study Reveals Human Factor Problems, Aviation Week & Space Technology, August 7, 1989.

[18] Dan Hughes and Michael Dornheim, Automated Cockpits Special Report, Parts I & II, Aviation Week & Space Technology, January 30-February 6, 1995

[19] Kathy Abbot, Presentation Slides, FAA Autoflight Mode Awareness Workshop, Seattle, WA, June 14-16, 1998.

[20] Commercial Aviation Safety Team, Final Report of the Loss of Control JSAT: Results and Analysis, Paul Russell and Jay Pardee, Co-chairs, December 15, 2000.

[21] Charles E. Billings. Aviation Automation: the Search for a Human Centered Approach, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 1997.

[22] Ricky W. Butler, Steven P. Miller, James N. Potts, and Victor A. Carreno, A Formal Methods Approach to the Analysis of Mode Confusion, in Proceedings of the 17th

AIAA/IEEE Digital Avionics Systems Conference, Bellevue, WA, October 1998.

[23] Judith Crow, Denis Javaux, and John Rushby, Models and Mechanized Methods that Integrate Human Factors into Automation Design, in Proceedings of the International Conference on Human-Computer Interaction in Aeronautics: HCI-Aero 2000, Toulouse, France, September, 2000.

[24] Steven P. Miller and James N. Potts, Detecting Mode Confusion Through Formal Analysis and Modeling, NASA Contractor Report NASA/CR-1999-208971, January 1999.

[25] John Rushby, Analyzing Cockpit Interfaces Using Formal Methods, Electronic Notes in Theoretical Computer Science, 43, URL: http://wwww.elsevier.nl/locate/entcs/volume43.html, 2001.

[26] John Rushby, Using Model Checking to Help Discover Mode Confusions and Other Automation Surprises, in the Proceedings of the 3rd Workshop on Human Error, Safety, and System Development (HESSD'99), Liege, Belgium, June 7-8, 1999.

[27] John Rushby, Judith Crow, and Everett Palmer, An Automated Method to Detect Potential Mode Confusion, in the Proceedings of the 18th AIAA/IEEE Digital Avionics Systems Conference (DASC), St. Louis, MO, October 1999.

[28] Jeffrey M. Thompson, Mats. P.E. Heimdahl, and Steven P. Miller, Specification Based Prototyping for Embedded Systems, in Proceedings of the Seventh ACM SIGSOFT Symposium on the Foundations on Software Engineering, LNCS, Number 1687, September 1999.

[29] Steven P. Miller, FGS Mode Confusion Visualization Report, NASA Contractor Report, March 2002.

[30] Nancy Leveson, Matts Heimdahl, Holly Hildreth, and Jon Reese, Requirements Specifications for Process-Control Systems, *IEEE Transactions on Software Engineering*, 20(9):684-707, September 1994.

[31] David Harel and Amnon Naamad, The STATEMATE Semantics of Statecharts, *ACM Transactions on Software Engineering and Methodology*, 5(4): 293-333, October, 1996.

[32] G. Berry and G. Gonthier, The Synchronous Programming Lanugage Esterel: design, semantics, and implementation, Science of Computer Programming, 19:87-152, 1992.

[33] Nancy G. Leveson, et. al., Safety Analysis of Air Traffic Control Upgrades, September 1997.

[34] Steven P. Miller and Alan C. Tribble, Taxonomy of Mode Confusion Sources, NASA Contractor Report, November 2001.

[35] Steven P. Miller, Alan C. Tribble, Timothy M. Carlson, and Eric J. Danielson, Flight Guidance System Requirements Specification Final Report, NASA Contractor Report, November 2001.

[36] Jeffrey Conklin, A Short Course in IBIS Methodology, available online at http://gdss.com/wp/IBIS.htm.