ELSEVIER

Reliability Engineering and System Safety 75 (2002) 133–145

RELIABILITY ENGINEERING & SYSTEM SAFETY

www.elsevier.com/locate/ress

Approaches to mitigating complexity-driven issues in commercial autoflight systems

Sanjay S. Vakil*, R. John Hansman Jr.

Massachusetts Institute of Technology, International Center for Air Transportation, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

Abstract

There appears to be broad consensus in the aviation community that increased automation in the cockpit has changed the task of flying commercial aircraft. The changes have been both beneficial, through the increase of capabilities and efficiencies, and detrimental, as indicated by accidents implicating automation as a contributory factor. It is hypothesized that the constraining factor on automation design has changed from technological to human. The evolutionary growth of the automation has increased complexity which is thought to have led to the lack of a global model of the automation upon which the training material and operator feedback can be designed.

Based on these analyses, three classes of complexity mitigation management techniques are explored. The first is to train pilots to understand and work within the current automation system. The second is to enhance feedback to allow more effective monitoring of aircraft systems, and to allow a reduction in the apparent order of the system to the pilot. Finally, a modified development process is suggested which explicitly considers the pilot in early design stages. It is believed that a process-oriented solution will be necessary for future automation systems. This process uses an explicit automation model as a basis for training material and for software requirement specification. © 2002 Published by Elsevier Science Ltd.

Keywords: Commercial autoflight syatem; Aviation community; Cockpit

1. Introduction and motivation

Advances in computation, algorithm, and sensor capabilities have driven a trend towards more automation in dynamical systems. In particular, the commercial aircraft cockpit has been augmented by automation causing changes to the task of flying an aircraft. Current advanced commercial transport aircraft, such as the Boeing B777/B747-400, and MD-11, and the Airbus A320/A330/A340, rely on autoflight systems (AFS) for flight management, trajectory control, and interaction with control surfaces. These systems have evolved from simple autopilots, to multiple processors capable of sophisticated and interrelated tasks. These tasks span the range from high-level flight management to low level control of individual actuators. Performance has been increased by allowing more accurate tracking of altitude and path targets, cost have been reduced by flying optimized fuel efficient paths, and alerting systems have been used to warn pilots or to deal directly with unsafe situations.

In theory, safety may be enhanced through automation by automating certain aspects of the flying task, or by augmenting the control characteristics to make the aircraft easier to fly. In practice, the rapid evolutionary development of advanced automation in commercial transport aircraft is suspected as a contributory factor in a number of incidents and accidents. Hull losses in which automation has been identified as a contributory factor occurred at Bangalore in 1990 [1], Strasbourg in 1992 [2,3], Nagoya in 1994 [4], and Cali in 1995 [5]. Numerous incidents have occurred, including a rapid pitch-up at Orly in 1994 [6], multiple incidents of overspeeds, and numerous large altitude deviations.

1.1. Accidents and incidents

As shown in Fig. 1, while the overall safety level has improved, the percentage of accidents attributed to human error has remained constant at approximately 70% [7]. What appears to be changing, however, is the nature of the errors. Previously, most accidents were caused by problems with the physical skills involved with flying the aircraft, or through errors of judgement. The new problems involve issues of management of the complex aircraft and associated automation systems. The role of the pilot has shifted from being a manipulator of the controls to be a manager of aircraft systems. Within the set of errors attributed to flight crews, automation problems are emerging as a key safety area. The incorporation of new flight automation has

---

* Corresponding author. Fax: +1-617-253-4196.
E-mail address: sanj@alum.mit.edu (S.S. Vakil).

0951-8320/02/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0951-8320(01)00090-4

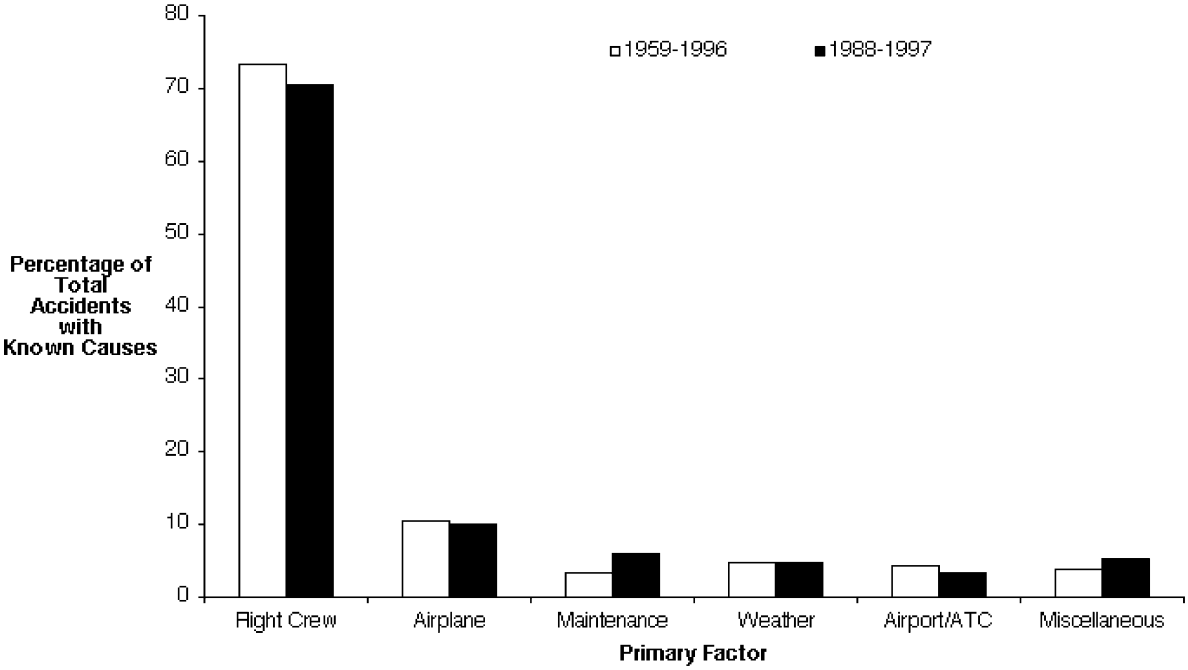


Fig. 1. Primary causes of aircraft accidents [7].

resulted in a new set of human factors issues. Sufficient concerns have been raised to warrant government investigation in the form of the *1996 FAA Report on the Interfaces Between Flightcrews and Modern Flight Deck Systems*. This document also discusses human factors and interface issues, which include mode awareness problems [8,9], incomplete pilot understanding of automation [10–13], and loss of automation situation awareness [14]. In contrast to mechanical aircraft failure, these problems appear to be based in confusion between the pilots' expectations of the autoflight system and what the system is actually doing.

1.1.1. Aviation safety reporting system

In order to further explore human factors issues and occurrence of automation issues, a search was performed on the aviation safety reporting system (ASRS) database [8,9]. The search covered the years 1990–1994 with a set of keywords designed to elicit problems related to mode awareness, a specific aspect of pilot-automation interaction. The keywords consisted of the following: annunciation, annunciator, FMC, flight management computer, FMS, flight management system, CDU, mode, capture, arm, automatic flight system, vertical, horizontal, and program. A total of three hundred ASRS reports were returned by the keyword search. After analysis, 184 were categorized as 'mode awareness problems'.

The most commonly reported errors were 'Programming Errors', 'Mode Transition Problems' and 'Insufficient Understanding of Automation'. It can be argued that dominance of the Programming Errors category may be overstated, since a single typographical error could cause an ASRS filing. However, if such a minor error can lead to a filing, it may be indicative of the more fundamental issue that the usage of automation can allow relatively minor errors on the part of the human to have significant repercussions.

The dominant causal areas are of particular importance because they suggest there can be confusion between the pilots' expectations of the automation and what it is actually doing. Mode transition problems indicate that pilots may not realize when the automation changes its behaviour or the implications of the new behaviour. Insufficient understanding of automation is equally problematic since it suggests that the pilots may not be able to supervise the automation: in order to effectively monitor automation, a pilot must understand what its intended behaviour should be.

As shown in Fig. 2, the reports were also categorized by the flight path component (vertical/speed, horizontal or both) that was impacted. Since the vertical flight path and the speed are implicitly coupled, they were grouped together. In Fig. 2, it can be seen that vertical/speed problems dominate many of the categories; of all categories 62.7% of the reports were of this type. In particular, the mode transition problems category is dominated by vertical/speed problems. The data classified into the insufficient understanding of automation also suggests a deficiency in knowledge of the vertical domain automation. It should be noted that due to the amnesty provision in ASRS submission there exists a potential for over-reporting vertical deviations. Air traffic control (ATC) radar can measure altitude much more precisely than location. This may be a cause for pilots to report vertical/speed incidents more often.

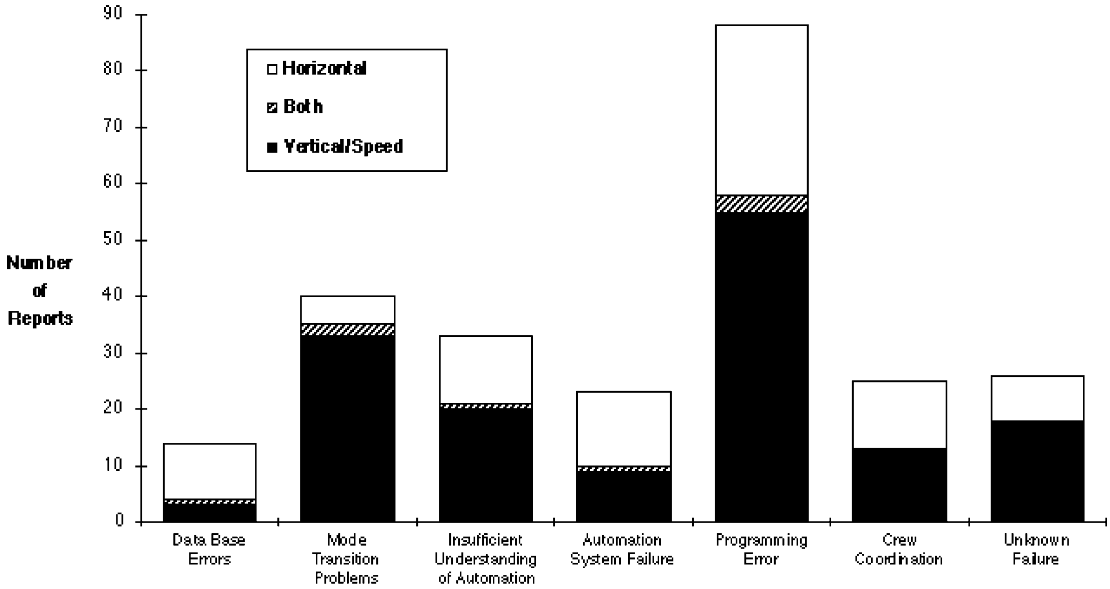


Fig. 2. Breakdown of ASRS reports into perceived causes and flight domain.

1.2. Development of cockpit automation

The overview of flight automation has suggested that the trend is to provide more automatic control over aircraft systems. Other researchers [13,15] have also shown that the total number of behaviours in a system can be very large and the transitions dependent on numerous conditional elements, increasing the perceived complexity of the automation.

1.2.1. Evolution of functions in automation

The number of functions available to pilots in aircraft automation have increased with time — typically, new functionality is added but older functionality is seldom removed. While it is difficult to gauge a specific measure of the number of functions across aircraft, a estimate was made of the number of independent ‘modes’ in a series of aircraft as shown in Fig. 3. Modes are defined as independent behaviours, which are supported by the aircraft in the vertical and horizontal components of flight. They correlate closely with the multiple state controllers in the aircraft automation. Data was derived from the open literature and training materials for multiple aircraft. As such, it may be incomplete from the standpoint of system design, but is a measure of the number of modes represented to pilots. In addition, it is difficult to compare across architectures: modes maybe undercounted based on how they are presented to the pilots. In particular, the high level Airbus PROF and Boeing VNAV modes consist of a set of submodes. These submodes are difficult to directly compare as the manufacturers have parsed the submodes differently. Therefore, these modes could not be counted separately. This implies both that the mode count in Fig. 3 is conservative for these aircraft and that the modes associated with trajectory control are underrepresented.

As an additional example, an estimate was made of the number of independent modes in a series of aircraft, based on the open literature and training materials for each aircraft

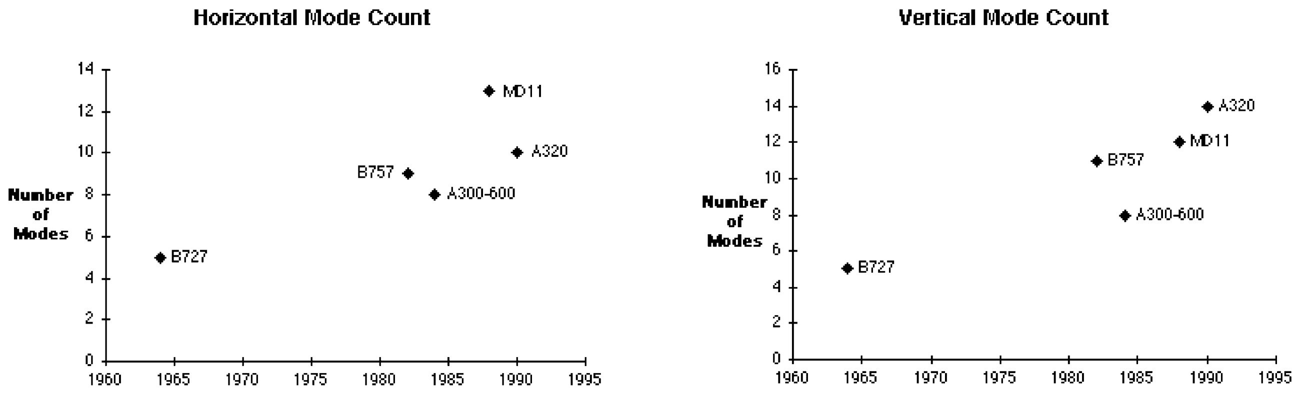
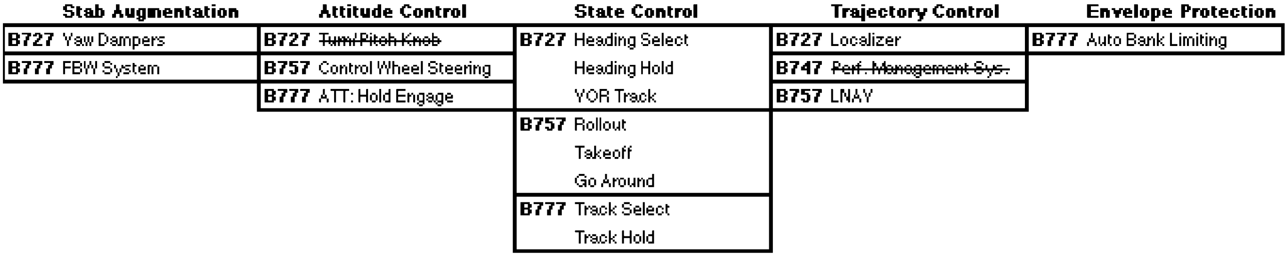


Fig. 3. Horizontal and vertical mode counts in selected aircraft.



Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
B727 Yaw Dampers	**B727** ~~Turn/Pitch Knob~~	**B727** Heading Select	**B727** Localizer	**B777** Auto Bank Limiting
B777 FBW System	**B757** Control Wheel Steering	Heading Hold	**B747** ~~Perf. Management Sys.~~	
	B777 ATT: Hold Engage	VOR Track	**B757** LNAV	
		B757 Rollout		
		Takeoff		
		Go Around		
		B777 Track Select		
		Track Hold		

Fig. 4. Lateral modes in the Boeing B777.

[16–22]. The data may be incomplete from the standpoint of system design, but is a measure of the number of modes articulated to pilots. As such, the number of modes may be undercounted. Figs. 4 and 5 show the growth of modes up to and including the B777. In these diagrams, all possible modes available through the autoflight system are shown, with those, which have been removed from previous generations shown struck out. Each mode is listed next to the generation of aircraft in which it was introduced.

2. Representation issues

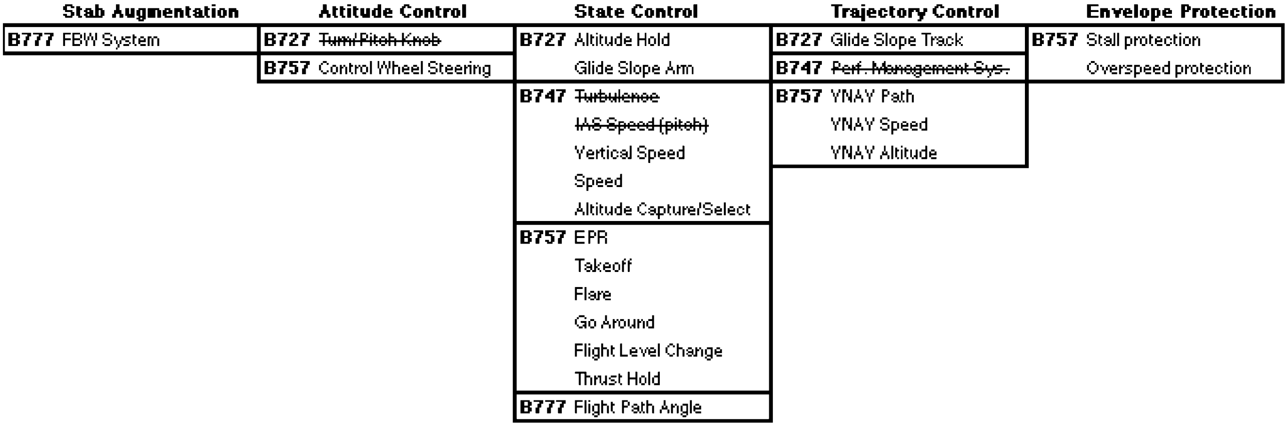
One concern with advanced cockpit aircraft is that the representation of the automation in may not be appropriate for pilots. This results in pilots developing ad hoc models of these systems in an experiential manner. In examining modern aircraft automation, and the ASRS data presented earlier it appears that mode transitions are important to the complexity of flight automation. Mode transition matrices provide a framework in which to examine and evaluate transitions and may provide insight into which modes require additional training.

2.1. Lack of appropriate representations of automation

A review conducted of existing flight automation systems suggests that the evolutionary development of these systems has resulted in a large and complex structure of operating modes which do not appear to have a simple, consistent, underlying global model [23]. Research was done through public information sources, such as aircraft manuals, focused interviews with line pilots and check airmen and direct contact with avionics manufacturers. This appears to be the case across all of the flight automation systems studied (B757, B767, B747-500, A320, A300, MD-11 and F-100). Avionics manufacturers who were contacted were not able to supply a functional model or logic/control diagram. The documentation presented to the FAA is a detailed specification of the implementation of the automation, but not an overall model.

Javaux [13] has put forward a mechanism based on spreading activation networks which accounts for the development of mental models during nominal conditions. *Frequential simplification* and *inferential simplification* are mechanisms by which the details of a sophisticated transition between modes can be reduced to a less comprehensive prototypical state.

Frequential simplification occurs when a sophisticated interaction is simplified during nominal operation by having many of the conditional elements satisfied. Inferential simplification creates an analytical basis for the incorrect application of consistency, when a pattern, which appears in some portion of the automation is incorrectly applied to another segment. It is difficult to distinguish which of these mechanisms is in effect, since they are related phenomena. The Airbus A300 hull loss that occurred at Nagoya may have been caused by a reasonable simplification of the autoflight system: that autopilot on the aircraft can be



Stab Augmentation	Attitude Control	State Control	Trajectory Control	Envelope Protection
B777 FBW System	**B727** ~~Turn/Pitch Knob~~	**B727** Altitude Hold	**B727** Glide Slope Track	**B757** Stall protection
	B757 Control Wheel Steering	Glide Slope Arm	**B747** ~~Perf. Management Sys.~~	Overspeed protection
		B747 ~~Turbulence~~	**B757** VNAV Path	
		~~IAS Speed (pitch)~~	VNAV Speed	
		Vertical Speed	VNAV Altitude	
		Speed		
		Altitude Capture/Select		
		B757 EPR		
		Takeoff		
		Flare		
		Go Around		
		Flight Level Change		
		Thrust Hold		
		B777 Flight Path Angle		

Fig. 5. Vertical/speed modes in the Boeing B777.

disengaged by forcing forward on the yoke with sufficient force. Unfortunately, this simplification is incomplete since the automation is not completely disengaged and maintains control over the trim of the horizontal stabilizer. This inaccurate simplification may have been a contributing factor to the accident as the crew appears to have been unaware that their commands were in conflict with the commanded state of the autoflight system.

2.1.1. Concerns with experientially developed mental models

In the absence of a simple, consistent and communicable model of flight automation, pilots appear to create their own models of the flight automation. These ad hoc mental models have several shortcomings. The most obvious of these is that the models may not accurately reflect the actual systems. Further, since these models are created independently by individual pilots, specific ad hoc models may not be accurate.

The basis of these models is grounded in both training material provided to the pilots and flight experience. The existing training material is based on a proceduralized, operational model with little causality or connection to the structure of the underlying system. It is expected that the actual mental models used by pilots are more sophisticated than those put forward during training and are influenced by their individual piloting background. These experiential models are also suspect, however, because they are created during nominal operations (where most experience occurs) and may not hold, or even become a liability, during emergency situations.

There are also some concerns about individual pilots reducing the complexity of the system, especially though the use of techniques which ignore parts of the system, conditional elements, or alternate paths through the automation. The learning which occurs during operation necessarily reinforces the modes, behaviour, and conditional elements which are seen most often [24]. Those which are experienced less often will be the ones which are removed from pilots' representations. A serious issue is that non-nominal or emergency modes are unlikely to be experienced directly with regularity. As such, these modes may become marginalized as a pilot has to deal with the burgeoning complexity of a system. Many automation behaviours exist with which pilots need to have a detailed understanding but will not occur regularly.

2.2. Representation of modal behaviour

Modes are a mechanism to allow disparate behaviours to coexist within a single system. Disparate behaviours will appear when new functionality is added to system, which cannot be parsed as an extension to existing function, or cannot be constructed by combining existing functions. In the case of autoflight systems, new modes were needed when necessary behaviours could not be generated by existing closed loop controllers. New modes were added in the form of new controllers. Dividing the system into separate controllers can allow selection between multiple behaviours. The active mode defines the active controller to determine the behaviour of the system.

Based on interviews, it appears that pilots model the system in a modal manner. If this is found to be a widely adopted representation, it may be the appropriate form upon which to base the training material. Focused interviews showed that pilots have adopted a modal representation of automation behaviour, as described by mode transition diagrams [25]. Note that this is different from a detailed Finite State Machine type of representation of the underlying automation, but rather an organization of the behaviour into separate modes. The differences appear in the parsing of what constitutes a mode, a trigger event, or a conditional clause. These differences acknowledge the operational viewpoint rather than the design viewpoint.

A formalism was developed to represent transitions between modes which is based on the formalism of finite state machines (FSMs). *Mode transition diagrams* are used to represent discrete elements of modal automation. FSMs are a standard tool used in the field of computer science to describe and design complex systems, including flight automation. Unlike FSMs, which are used by engineers during design and analysis, mode transition diagrams describe the structure of the automation as experienced by the pilot. FSMs consist of a set of states, a set of transitions between states and the criteria, which cause transitions to occur. Modal automation systems can be represented using the same notation, which is used for finite state machines. In this nomenclature, the states correspond to the modes of the automation, transitions move between modes, and the transition criteria consist of the conditions, which must be satisfied. The analogous diagrams are termed mode diagrams and the matrices, which are derived, are called mode transition matrices [26].

Fig. 6 shows the modal structure of a simple autopilot. There are a total of six modes in this diagram: horizontal autopilot off (HOFF), localizer track (LOC), vertical autopilot off (VOFF), heading track (HDG) vertical speed (VS), and glideslope track (GS). In this example, HOFF can transition to LOC or HDG, but not to VOFF, GS or VS.

Mode	HOFF	LOC	VOFF	HDG	VS	GS
HOFF		1		1		
LOC	1			1		
VOFF					1	1
HDG	1	1				
VS			1			1
GS			1		1	

Fig. 6. Transition matrix for simple autopilot system.

VOFF can be transitioned to from VS or GS, and so on. The axes denote the possible states, and the matrix elements correspond to whether a transition is possible or allowed between two different modes. Each row i is the set of transitions which leave mode i. The column j in row i has an entry corresponding to whether a transition exists from mode i to state j. If the mode transition matrix is some matrix $\mathbf{T}$, then T_{ij} is equal to one if a transition exists between states i and j.

Rather than matrix cells simply showing ‘allowable’ transitions, they could alternately be populated with other important characteristics of transitions such as the conditional upon which the transition is predicated or how often the transition occurs. This makes a mode transition matrix a mechanism to identify those modes that are rarely used and may be susceptible to inappropriate frequential or inferential simplification.

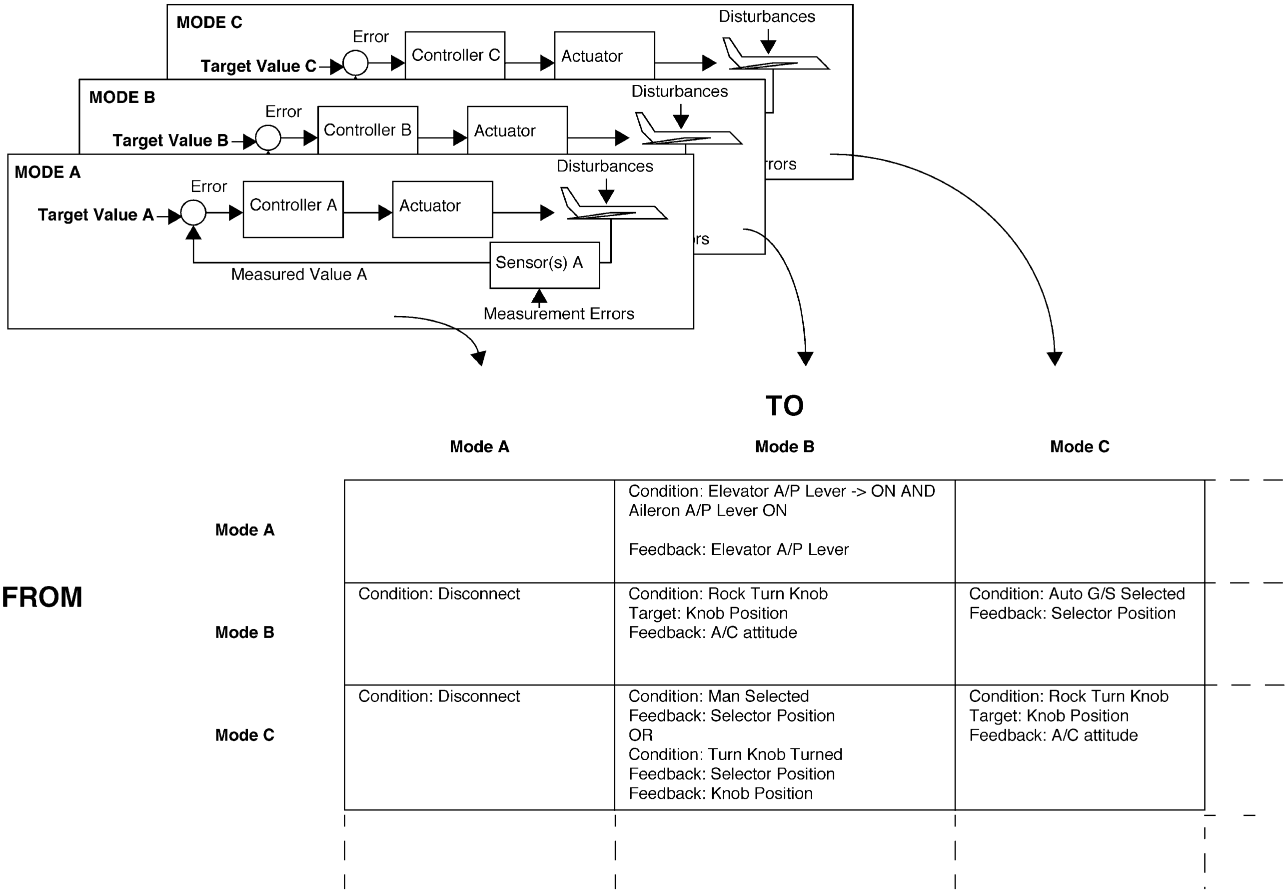
2.2.1. Hybrid automation representation

Hybrid automation representation extends the idea of mode transition matrices to capture both the quasi-steady-state and the discrete behaviour of aircraft automation systems. The former can be completely modelled using the engineering representation of control block diagrams at various levels of loop closure. The discrete behaviour can modelled using mode transition diagrams and matrices. The representation integrates the continuous representation of control block diagrams with mode transition matrices and diagrams [26] for discrete representation. Fig. 7 shows the major elements of this model. To read this diagram, the ‘From’ modes are shown in the rows and then ‘To’ modes are listed in the columns. The transition from Mode A to B is shown to occur when the Elevator Autopilot Lever is in the ON position under the condition that the Aileron Autopilot is also in the ON position. The feedback to the pilot consists of the position of the lever itself.

2.2.2. Analysis of automation using mode transition matrices

An issue with experientially developed models is that pilot interaction with modes which are rarely used or modes which are used regularly but for short duration will lead to incomplete or inappropriate mental models. These modes may not be used sufficiently to populate whatever mental representation the pilot is using. In the case of modes of limited duration, it is unlikely that there is time to consider or model the effects of inputs to the system, simply because little time exists to provide these inputs [27].

In a larger sense, this problem exists in many aspects of aviation. Much of standard and recurrent training deals with



TO

		Mode A	Mode B	Mode C
FROM	Mode A		Condition: Elevator A/P Lever -> ON AND Aileron A/P Lever ON Feedback: Elevator A/P Lever	
	Mode B	Condition: Disconnect	Condition: Rock Turn Knob Target: Knob Position Feedback: A/C attitude	Condition: Auto G/S Selected Feedback: Selector Position
	Mode C	Condition: Disconnect	Condition: Man Selected Feedback: Selector Position OR Condition: Turn Knob Turned Feedback: Selector Position Feedback: Knob Position	Condition: Rock Turn Knob Target: Knob Position Feedback: A/C attitude

Fig. 7. Hybrid automation representation.

	Takeoff Heading Hold	VNAV Heading Hold	VNAV Heading Select	VNAV Spd Int LNAV	VNAV LNAV	VNAV Des Now LNAV	Vertical Speed LNAV	Flight Level Ch LNAV	Flight Level Ch Heading Select	Altitude Hold LNAV	Altitude Hold Heading Select	Flight Level Ch Localizer	Glide Slope Localizer
Takeoff Heading Hold		4%	6%										
VNAV Heading Hold			3%										
VNAV Heading Select					11%								
VNAV Spd Int LNAV					2%								
VNAV LNAV			4%	2%		4%	2%	4%	2%	2%	2%		
VNAV Des Now LNAV					3%								
Vertical Speed LNAV								2%					
Flight Level Ch LNAV					2%				5%				
Flight Level Ch Heading Select								2%			5%	2%	3%
Altitude Hold LNAV									2%				
Altitude Hold Heading Select									3%				2%
Flight Level Ch Localizer													
Glide Slope Localizer													

Fig. 8. Observed autoflight transition matrix showing absolute percentage of transitions.

situations which are rare: engine fires, tire blowouts, go-arounds, etc. These are dealt with by creating and reinforcing procedural behaviours during simulator training. Similar training and effort is not spent reinforcing models of the automation itself.

Mode transition matrices can be used to help in organizing and identifying which modes and which transitions suffer from an insufficient experiential basis. Since the mental representations used by pilots appear to be experientially based, the most benefit may be gained by tailoring the automation training to individuals. This may be possible through access to pilot-specific data from data recorders.

Fig. 8 is a mode transition matrix adapted from work by Degani [28] observing mode transitions in operational, revenue-earning flights of a Boeing 757. In this matrix, rather than populating cells where transitions are possible, the cells are populated with the absolute frequency of observed transitions. In contrast with earlier examples, the aircraft autoflight modes include both vertical and lateral behaviours, so that a single mode is defined as the behaviour of the system in both axes. Mode transitions which were observed with a frequency of less than 2% are not included on the diagram.

In addition to identifying the modes and transitions which are not used often, there are several additional insights to be gained from the matrix. The first is that certain modes were only observed to transition to a single other mode. For example VNAV-heading hold was only observed to transition to VNAV-heading select. This does not imply that there is only allowable transition from VNAV-heading hold, but does highlight how the modes are used operationally. Since the matrix is based on a finite set of data, all possible transitions were unlikely to be observed. By accessing quick access data recorder information on commercial flights it should be possible to identify common transitions and those, which are rarely used.

Other modes have no observed exit transitions at all. The flight level change-localizer and glide slope-localizer modes are terminal modes, as can be seen from the empty row in the transition matrix. While there are allowable transitions from these modes (usually to go around mode) they were not observed. Large blank areas can be observed in the matrix, corresponding to sets of transitions, which were not observed. In many instances, this is due to certain modes being highly specialized and so of limited usage. An example is takeoff-heading hold, which only used during takeoff and climbout and not later in flight.

Columns and rows which are highly populated represent the modes in which the aircraft automation spends the bulk of its time. In this data VNAV–LNAV has a large number of transitions both in and out. In particular, note the most frequent transition between VNAV-heading select and VNAV–LNAV. This is hypothesized to be due to vector-based air traffic control transitioning to flying a predetermined flightplan.

2.3. Cyclomatic complexity

Mode transition matrices provide a macroscopic of the modal behaviour of a system in that it does not examine the details of individual mode transitions. Cyclomatic complexity is an analysis tool to probe the characteristics of individual transitions between modes.

Cyclomatic complexity is an analysis technique originally used to examine the complexity of structured software written on mainframe computers [29]. In the analysis, cyclomatic complexity determines the number of linearly independent paths through the system. The original goal was to examine the complexity associated with multiple branching code modules or states to gain insight into the impact of structure programming. This approach has been extended to examine the complexity of any system, which can be shown as a linked set of edges and nodes. A node is some type of state or decision point within a system, and an edge is a mechanism to connect nodes. In particular, it was used in the examination of an autoflight mode transition. This is based on the premise that determining whether a transition will occur is dependent on the evaluation of a set of predicating conditions and is analogous to how a structured program is dependent on branching decisions to determine its flow of control.

Further, cyclomatic complexity appears to be a useful analysis tool to examine transition characteristics which are hypothesized to impact the apparent or perceived complexity of autoflight automation. In order to monitor an autoflight system, a pilot needs to be able to track the evolution of the state of the automation in addition to the state of the aircraft dynamics. Since automation can directly control the behaviour of the aircraft, the state of the automation needs to be understood in order to predict the future aircraft state and to detect when it is inconsistent with what was intended or expected. In order to do this, a pilot must have a representation of the automation itself in order to monitor conformance.

Cyclomatic complexity is a rationale approach to analyzing autoflight mode transitions which counts the number of linearly independent paths. Each path corresponds to a set of evaluations, which must be made by a pilot in order to ascertain the future state of the system. Cyclomatic complexity is dependent on the number and structure of the conditional elements in the transitions and is hypothesized to be useful in the analysis of the apparent or perceived complexity of a system. In particular, autoflight mode transitions are thought to have their apparent complexity impacted by the number and structure of conditional elements [25]. These characteristics correspond to those which are identified in the mode transition diagram and which are hypothesized to have an impact on the apparent complexity. Fig. 9 shows these elements. The starting and ending mode are necessary to identify the transition and the total number of modes is hypothesized to impact the system complexity and can be analyzed from the size of the mode transition matrix.

Another factor which appears to effect complexity is the number of transitions between modes and the number of different new target values which can be specified by the transition. Recall that the behaviour of a system is defined both by the active mode and its target value. As such, each transition path, which results in a new target value is considered independently. As an example, in Fig. 9, there

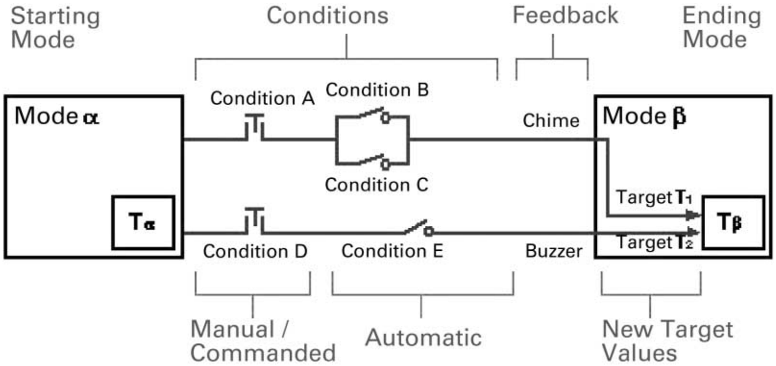


Fig. 9. Mode transition diagram abstraction.

are two transitions paths, corresponding to the new target values T_1 and T_2. This also allows the representation to describe staying in one mode while changing target values. Finally, the number and structure of the conditions are identified, as they can suppress transitions. Feedback that is provided for the transition is a tool to allow the human operator to monitor the transitions as they occur — it is a mechanism to mitigate the effects of complexity, and is not thought to directly impact the implicit or structural complexity of the transition at a given level of abstraction.

Rasmussen [30] and others [31] discuss the necessity to effectively interact at a knowledge-based level. At this level of understanding, the pilot must have a model of the automation which can be cognitively ‘run’ in order to predict future aircraft states. Multiple possible outcomes are generated based on this predictive analysis and the most likely of these outcomes is selected. The complexity of this model has an impact on the perceived complexity of the automation. Cyclomatic complexity is a tool to analyze the structure of the model utilized by the pilot in the process of monitoring the automation. Each linearly independent path through the autoflight systems corresponds to a set of evaluations in the process of monitoring.

3. Approaches to mitigating complexity issues

This section will examine approaches to managing and mitigating the complexity of existing automation systems. Pre-emptive approaches have three classes, corresponding to the time and cost of implementation. The first is to apply more effective training with knowledge of the structure of the automation as well as known aircraft problems [32]. The second is to critically examine the feedback provided by the automation and enhance it to allow the system to become more observable, especially in light of hidden conditional elements and targets. Ultimately, however, the process by which these systems are designed must be examined. A process is presented which is designed to create systems, which may not require the same amount of simplification by pilots.

3.1. Training and procedure modifications

Aircraft accidents typically occur early in the lifetime of a particular aircraft system. Fig. 10 shows the Hull Loss Accident Rate of the worldwide commercial aircraft fleet from 1988–1997 by individual aircraft [7]. Multiple aircraft that have been introduced since 1981 are shown with the accident rate per one million departures. While noting that the statistical significance of the information in this chart is limited since hull loss accidents are rare events, there is a significant difference between the introduction of the Airbus A319/320/321 series as compares to other aircraft. Part of the reason behind the anomalous nature of the A320 derivative record is an early hull loss during operational usage at the Habsheim airshow on 26 June 1988. The basis for these early problems, which are not limited to the A320 series, may be the identification and understanding of emergent behaviours of the aircraft through operational usage. As more experience is gained with new aircraft systems, fewer accidents occur.

The accident rate of this aircraft also appears to be improving as pilots and airlines gain more experience with its detailed behaviour, and as these details are disseminated. A hypothesis for this improvement is that training material, procedures, and flight crews are becoming more proficient with the aircraft. This is consistent with the nature of the A320, which was the first fully digital commercial fly-by-wire aircraft. It also included numerous departures from previous designs, such as a full authority envelope protection system, a side-stick controller, and non-moving throttles. This experience also explains the lack of hull losses of the recently introduced A330 and A340, which have very similar cockpit automation.

If the behaviour of automation is found to be confusing or insufficiently documented, especially in the aftermath of an accident or incident, flight crews of affected aircraft fleets are notified. The notification process may be initiated by the manufacturer or by airlines directly, and often consists of additions or modifications to the flight crew operating manuals. This information is also incorporated into training and changed procedures.

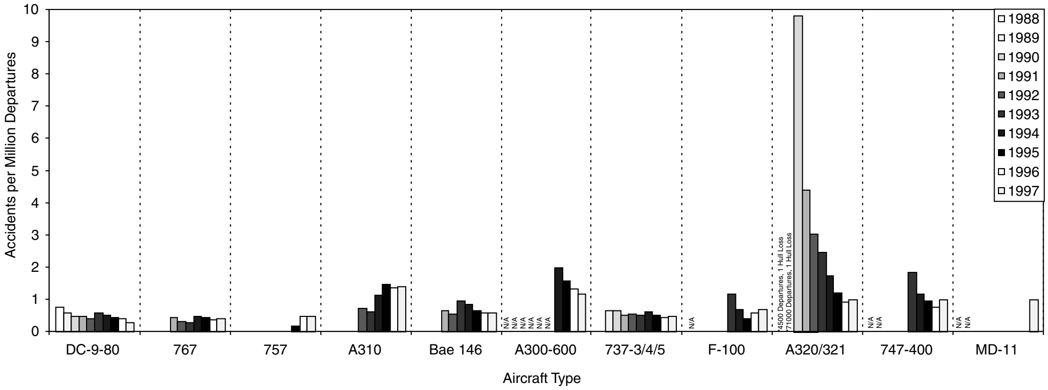


Fig. 10. Worldwide Hull Loss Accident Rates (1988–1997) [7].

As an example, in the aftermath of the hull loss in Cali, Colombia, pilots flying the Boeing B757/767 were sent a bulletin (Boeing 757/767 Operating Manual Bulletin No. 757/767-19) which described an updated procedure to use during route modification. The updated procedure was an explicit work-around for the behaviour of the aircraft automation.

3.2. Feedback

Another approach to mitigating complexity issues is to provide sufficient information to the pilot to accurately track the state of the automation. If this is not done, then using the parlance of control engineers, the system is not 'observable'. Accurately characterizing necessary automation feedback can support modifications that make the system more tractable. However, since such changes to aircraft are much more rare, and expensive, than changes to training or procedures, they are likely to only be done in extreme situations. An example of such a situation is the retrofit to the A320 to call more explicit attention to the distinction between being in flight path angle mode and vertical speed mode. This change was made after an accident which was contributed to by confusion between these two modes [3].

Feedback can also be used to serve to call attention to transitions or conditions which are operationally rare. If a rare mode has been entered (such as an envelope protection mode), or a rare condition has caused an unexpected transition, this can be made clear to the pilot. There is a trade-off between how often a particular transition is made and how familiar a pilot may be with it. By selectively drawing attention to events which are 'rare', it may be possible to compensate for an experiential liability [33].

At a more fundamental level, the purpose of feedback is to change the nature of the system as experienced and interacted with by the pilot. In a manner similar to procedure design, a simpler system can be created by reducing the interaction to a smaller set of more capable behaviours. Alternately, a system can be made easier to monitor by presenting a pilot which more useful feedback about the state of the aircraft. Based on the vertical domain dominating problems with mode transitions in both the ASRS review and the web survey, an electronic vertical situation display was prototyped and evaluated [9].

3.2.1. Prototype electronic vertical situation display

An example of enhanced feedback is a prototype electronic vertical situation display, shown in Fig. 11, is analogous to the moving map display, but depicting the vertical progress of the aircraft. The display has four distinct areas. At the top of the display is the mode display window, showing the current and anticipated modes, control allocations and target states. At the left is a scalable altitude tape. The bottom window can either display the path distance (if in LNAV mode), or the range directly ahead of the aircraft. Finally, the main window shows the aircraft vertically in relation to the upcoming waypoints and mode transition points.

The current mode of the automation needs to be identified along with any of the specific attributes of the mode such as target values and control allocation. In Fig. 11, the current mode is identified in the top window in green text, directly above the aircraft symbol. In this example, the aircraft is in VNAV path descent (VPATH). An example of a transition criterion is the dashed magenta line at 15,000 ft, which is the altitude dialed into the mode control panel.

Anticipated modes consist of the future modes into which the automation expects the aircraft to transition. On the EVSD, the anticipated mode is shown in the top window above the point where it is predicted to be engaged. The anticipated targets and control allocations are depicted in a manner similar to the current mode. In Fig. 11, the system is predicting a speed violation and a mode transition to the VNAV speed mode (VSPD). In this mode, the display shows that the vertical path will be controlled by the throttles to Idle, and the speed will be controlled by elevators to 320 kts. Note that both the target states and the control allocation change when the new mode is engaged. Another mode change is anticipated once the aircraft reaches the MCP altitude of 14,000 ft, approximately 12 nmi ahead of the aircraft. The aircraft will switch to VPATH. Once the automation switches to this mode, the altitude becomes a target, as shown in the box underneath the VPATH text.

This display has a green 'path predictor' line which shows the future vertical state of the aircraft using a linear extrapolation based on the current automation state. This line shows the behaviour of the aircraft in the context of the impending airspace and how it may differ from what is expected by the pilot. The feedback provided is much more useful that what currently exists in aircraft in helping to track conformances with commands.

The prototype electronic vertical situation display was evaluated through a series of part-task simulation-based scenarios. The display was found to significantly improve mode awareness understanding and the detection of mode

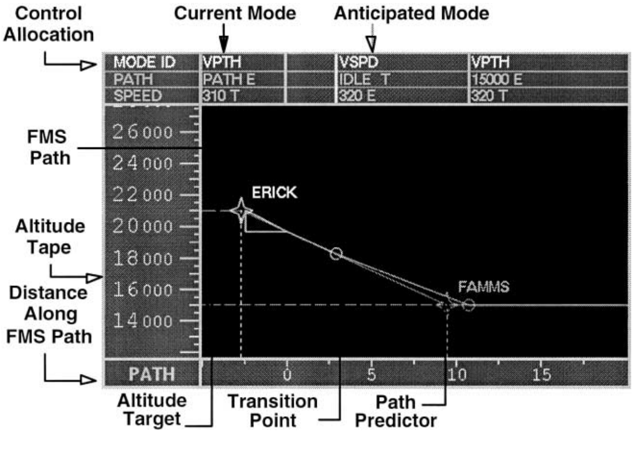


Fig. 11. Prototype electronic vertical situation display.

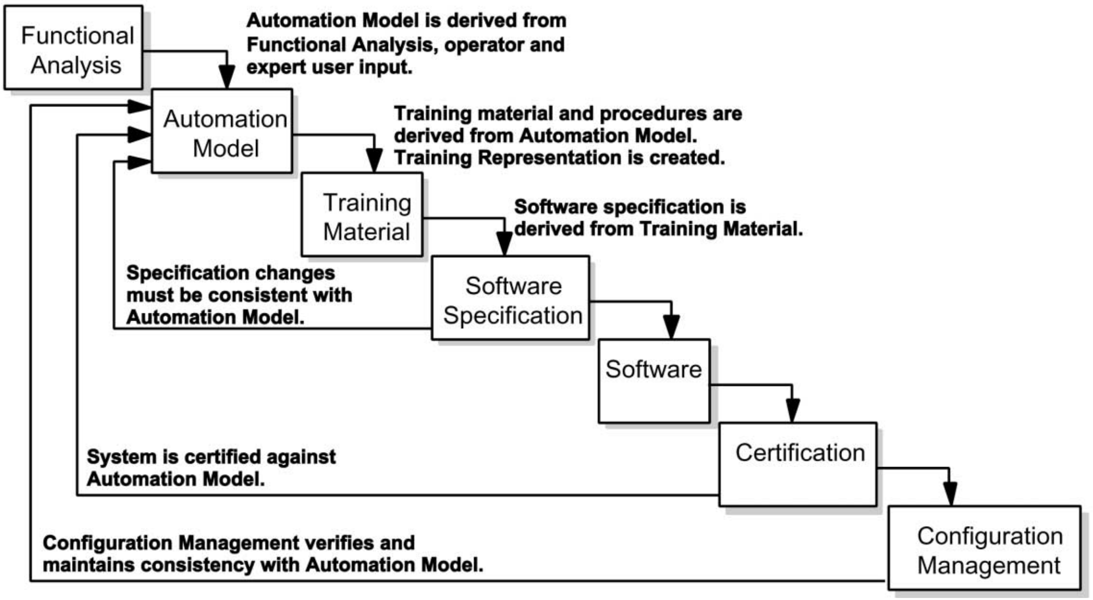


Fig. 12. Operator directed process (Waterfall model).

awareness problems in both subjective and objective measures of subject response. Objective results were particularly strong when the anticipation functions of the EVSD could be used to foresee an event before it actually occurred. In addition, subjects were much more specific when reporting problems to air traffic control. For example, rather than simply reporting that they were unable to make a crossing restriction, subjects would also report how far past of the waypoint the altitude would be acquired. Several subjects also mentioned the additional utility of having a vertical image of the aircraft's programmed flight.

3.3. Process-oriented solutions

The most fundamental solutions to automation misunderstanding is to develop automation in a manner to be more consistent with the an operator's conceptual model of the system. The costs and development times involved make this unlikely to occur as retrofits to existing aircraft since there is already a large installed equipment base. More likely, this solution will only be undertaken when new functionality in the cockpit requires redesign in future aircraft. Work has been underway to determine if guidance can be provided a priori to designers to allow the creation of less vulnerable systems. One approach which should be consisted is a process oriented solution: the 'operator directed process'.

3.3.1. Operator directed process

Many of the problems which have appeared in modern aircraft appear to be related to a mismatch between engineering and pilot models. The issue is of a lack of consistency, especially among rarely used modes. One solution is to explicitly consider the human operators early in the design process to prevent these mismatches. This also assists in capturing the limitations that the operator may place on the autoflight automation and to constrain system complexity early in the design process. The fundamental principle is to increase system usability through the constraint of complexity by articulating an operationally appropriate model of automation for use by operators. This may also be extended to assist in certification process.

In order to support the development and certification of complex automation systems that consider flight crew operational understanding, the use of an operator directed process (ODP) is proposed. The ODP is shown schematically in Fig. 12. The major difference in this process is that the training material is the source of the system specification rather than vice versa. Developing training material early forces consideration fundamental issues in human–machine interaction early in the process. This contrasts with existing development cycles that use training material to *document* system design. The intent is to develop a less error-prone and more understandable system by requiring consistency between the training material, procedural usage, and the software, and by limiting the complexity of the system through the articulation of a model for the operator. This enables the explicit consideration of the human operator early in the development process.

The ODP is based on earlier ideas, including user centered design [31], knowledge-based interface design [34], and others [35]. The process is particularly appropriate to aircraft systems because of the skilled set of operators (i.e. pilots) who may have a differing characterization of tasks than designers. Another major factor is the manner in which procedures influence the task of flying by imposing external structure. For example, in the case of standard instrument departures (SIDs), standard terminal arrival routes (STARs), and automated approaches, the automation is tied to the structure of the procedure, and the task fundamentally requires the use of the automated system. By contrast, much of the research that has been done into how humans interact with computers has used case studies where operator skill is tied to the use of the computer, rather than the larger task [34,35], and where fewer operational impositions exist.

The existing development processes for flight automation were developed in an era where computing power was at a premium and the capabilities of computing systems limited. The shift to a new development process is justified based on the flexibility, capability and complexity afforded by modern processors and software systems. One of the goals of the ODP is to constrain the complexity of these systems to a level which human operators can internalize and understand while maintaining the necessary functionality. It should be noted that the ODP might suggest limiting functionality for some systems if the required system is so complex that it proves intractable to the pilot.

What is necessary is a mechanism with which to capture the most important system elements, as found in the 'engineer's' representation of the system in a form which is suitable for the operator. This is a difficult task since is requires capturing a fundamentally complex task in a simpler form. An appropriate model for the operator is likely to be one that is rooted in operational domain and acknowledges the background of the user. The representational form of this model is dependent on the automation it is attempting to describe. A number of possible modeling bases are presented in Fig. 13 (see also Ref. [36]).

In Fig. 12, the ODP is shown to follow the 'waterfall model' used in classic software engineering. The waterfall model flows information and design considerations 'downstream' to be dealt with by the next stage. The major stages of this process are needs analysis and specification, design, implementation, testing, and maintenance and upgrades. This is used as an explanatory diagram in order to show dependencies. In practice, it is closer to Boehm's 'spiral' model which consists of a series of repeating stages of iteration, where updates are made to an operational prototype of the final system [39]. The feedback loops are shown as the revers ('upstream') arrows which provide input to the automation model.

Control Block Diagrams	Control block diagrams are useful for continuous systems where they can accurate represent the continuous behaviour of a mode. Typically they are used by system designers.
Procedural Constructs	Procedures are used extensively in Operating Manuals and provide a well-defined procedure to accurately instruct the automation. However, they can become confusing: "Through the FCU, an immediate climb/descent is initiated by selecting the desired altitude in the ALT SEL window and either pulling the set knob or pressing the LVL/CH P/B to engage the LVL CHANGE mode. Pressing the LVL/CH P/B also disengages PROFILE, however, if PROFILE is engaged, pulling the set knob does not disengage it, rather it initiates an immediate climb/descent to the altitude selected on the FCU. The exceptions are..."
Finite State Representations and Variants	Use and extend Finite State Machine notation, terminology and analysis techniques to gain insight into underlying modal structure of complex systems.
Analogical Descriptions	Many systems are described by an analogy to a previously understood description. An example might be that "This is controlled just like a B727 autoflight system". Graphical user interfaces in modern computers use a desktop metaphor. Spreadsheets embrace and extend the ledger book paradigm.
Anthropomorphic Descriptions	Automation can be designed to emulate a human agent. If successful, the operator can interact with the automation with very little training. However, this approach is limited by language accuracy, completeness, and ambiguity.
Linguistic Descriptions	Used by Riley ([37], [38]) to build system functionality with a consistent language to describe Air Traffic Control Directives. This is a wrapper around the existing automation and its issues. There is some concern that functionality and specificity may be lost with these high level descriptions.
Petri Nets	Useful for capturing all permutations of interactions and for capturing details of reactive systems.
Explanatory Descriptions	Purely explanatory descriptions are used extensively in Operating Manuals to provide a template for usage. An example from a current manual is shown below: "For demonstration purposes, assume level at FL330, a climb to FL370 is desired at PPOS. A manual climb to FL370, utilitizing the PROG page to input data to the FMS is affected in the following manner: • Write 370 or FL370 in the SP • Press LSK • Selected 370 in the ALT SEL window of the FCU"

Fig. 13. Possible automation model representations.

4. Conclusions

This paper has presented an overview of problems which have been found in modern cockpits which may be attributed to the complex state of aircraft automation. Multiple possible causal factors were suggested for this state, including incremental development of systems, the addition of more functionality, and the growth of possible targets given the increase in measurable states. The effect of this complexity is the existence of aircraft systems, which may be becoming too complicated to be adequately understood by pilots. These problems are further complicated by the lack of a complete and straightforward representation of the automation suitable for the design of training material and, therefore, the mental representations of pilots. In response, pilots develop their own ad hoc models of automation based on experiential interaction. These models may be simplified in specific ways to create a tractable system.

The aviation automation problem is currently addressed by existing mechanisms, albeit at a high cost. Issues found during operational experience are typically resolved through training. For future systems, approaches which enhance feedback to enable observation of the state of the automation are expected. The larger concern is for aerospace systems, which introduce new capabilities and non-aerospace systems. For these it is becoming apparent that the process by which automation systems are designed must be changed. The Operator Directed Process suggests a, but not the only, solution.

Acknowledgements

This work was supported by NASA Grants NAG-1-1581 and NAG-1-1857 and by FAA and NASA Grant FAA-95-G-017.

References

[1] Lenorovitz J. Indian A320 crash probe data show crew improperly configured aircraft. Aviat Week Space Technol 1995;January 30.

[2] Hughes D. Accidents direct focus on cockpit automation. Aviat Week Space Technol 1995;January 30.

[3] Hughes D. Incidents reveal mode confusion. Aviat Week Space Technol 1995;January 30.

[4] Mecham M. Autopilot go-around key to CAL crash. Aviat Week Space Technol 1994;May 9.

[5] Federal Aviation Administration. FAA Report on the interfaces between flightcrews and modern flight deck systems, 1996.

[6] Sparaco P. A310 inquiry target FCU. Aviat Week Space Technol 1994;October 10.

[7] Boeing Data. Hull Loss Accident Rates 1959–1997. Boeing Commercial Aircraft Company, Seattle, Washington.

[8] Vakil S, Hansman RJ, Midkiff A. Feedback mechanisms to improve mode awareness in advanced autoflight systems. Eighth International Symposium on Aviation Psychology, 1995.

[9] Vakil S, Hansman RJ, Midkiff A. Impact of vertical situation information on vertical mode awareness in advanced autoflight systems. 14th AIAA/IEEE Digital Avionics Systems Conference, 1995.

[10] Sarter N, Woods D. Pilot interaction with cockpit automation: operational experiences with the flight management system. Int J Aviat Psychol 1992;2(4):303–21.

[11] Weiner E, Nagel D, editors. Human factors in aviation. New York: Academic Press, 1988.

[12] Vakil S, Hansman RJ. Development and evaluation of an electronic vertical situation display. MIT Aeronautical Systems Laboratory Report, ASL-96-2, 1996.

[13] Javaux D, De Keyser V. The cognitive complexity of pilot-mode interaction. A possible explanation of Sarter and Woods's classical results. In: Boy G, Graeber C, editors. Proceedings of the International Conference on Human–Computer Interaction in Aeronautics, 27–29 May, Montreal, Canada, 1998.

[14] Endsley M. Towards a theory of situational awareness in dynamics systems. Human Factors 1995;37(1):32–64.

[15] Sherry L, Polson P. Shared models of flight management system vertical guidance. Int J Aviat Psychol 1999;9(2):139–53.

[16] American Airlines. Boeing 727 Operations Manual, 1997.

[17] Boeing. Boeing 737-200 Operations Manual, 1985.

[18] Boeing. Boeing 747-200 Operations Manual, 1985.

[19] Boeing. Boeing 757 Operations Manual, 1986.

[20] Boeing. Boeing 737-300/-400/-500 Operations Manual, 1989.

[21] Boeing. Boeing 767-200 Operations Manual, 1993.

[22] Boeing. Boeing 777-200 Operations Manual, 1997.

[23] Vakil S, Hansman RJ, Midkiff A. Preliminary results of the experimental evaluation of a prototype electronic vertical situation display. Eighth European Aviation Safety Seminar, Amsterdam, The Netherlands, 1996.

[24] Johnson-Laird PN. Mental models. Cambridge, MA: Harvard University Press, 1983.

[25] Javaux D. Personal Communication, 1999.

[26] Vakil S, Hansman RJ. Understanding automation structure by analyzing mode transition matrices. 17th Digital Avionics Systems Conference, Seattle, USA, 1998.

[27] Palmer E. Oops, it didn't arm — a case study of two automation surprises. In: Jenson R, Rakovan L, editors. Proceedings of the Ninth Symposium on Aviation Psychology, Columbus, Ohio, 1996.

[28] Degani A. Modelling human–machine errors: on modes, error and patterns of interaction. Doctoral Thesis, Atlanta, GA: Georgia Institute of Technology, 1994.

[29] McCabe T. A complexity measure. IEEE Trans Software Engng 1976;2(4):308–20.

[30] Rasmussen J. Information processing and human–machine interaction: an approach to cognitive engineering. New York: North-Holland, 1986.

[31] Norman D. The design of everyday things. New York: Basic Books, 1989.

[32] Weiner E, Chute R, Moses J. Transition to glass: pilot training for high technology transport aircraft. NASA Contractor Report, NASA/CR-1999-208784, 1999.

[33] Tognazzini B. Tog on interface. Reading, MA: Addison-Wesley, 1992.

[34] Shneiderman B. Designing the user interface: strategies for effective human–computer interaction. Reading, MA: Addison-Wesley, 1987.

[35] Card S, Newell A, Moran T. The psychology of human–computer interaction. Hillsdale, NJ: Erlbaum, 1983.

[36] Rouse WB, et al. The role of mental models in team performance in complex systems. IEEE Trans Syst, Man, Cyber 1992;22(6):1296–308.

[37] Riley V. Presentation at the Massachusetts Institute of Technology, 1995.

[38] Riley V, DeMers B, Misiak C. The Cockpit Control Language, 1997.

[39] Boehm BW. Software engineering economics. Englewood Cliffs, NJ: Prentice-Hall, 1981.