From *International Conference on Human-Computer Interaction in Aeronautics: HCI-Aero 2000*, pp. 163–168, Toulouse, France, September 2000.

Models and Mechanized Methods that Integrate Human Factors into Automation Design

Judith Crow
Computer Science Laboratory
SRI International
Menlo Park CA 94025, USA
crow@csl.sri.com

Denis Javaux
University of Liège
Work Psychology Dept
FAPSE-ULG. Bat. B-32
4000 Sart-Tilman, Belgium
d.javaux@ulg.ac.be

John Rushby
Computer Science Laboratory
SRI International
Menlo Park CA 94025, USA
rushby@csl.sri.com

ABSTRACT
Recent work has shown a convergence between the Human Factors and Formal Methods communities that opens promising new directions for collaborative work in calculating, predicting, and analyzing the behavior of complex aeronautical systems and their operators. Previously it has been shown that fully automatic, finite-state verification techniques can be used to identify likely sources of mode confusion in existing systems; in this paper we focus on use of these techniques in the design of new systems. We use a simple example to demonstrate how automated finite-state techniques can be used to explore autopilot design options, and then suggest additional applications for this technique, including the validation of empirically-derived, minimal mental models of autopilot behavior.

KEYWORDS
Human factors, formal methods, finite-state machines, automation, design

MODELS, METHODS, AND MECHANIZATION

Research in aviation psychology and in human factors (HF) has provided valuable insights and effective methods for evaluating human-computer interfaces and systems in modern aircraft. However, these methods are generally empirical and a posteriori: they rely on questionnaires (e.g., [17,19]), simulator experiments (e.g., [17,18,19]), or reporting systems (e.g., [22]), and can therefore be used only when the aircraft, a prototype, or a prototype simulator is available. These methods are descriptive rather than analytic, so that application of HF knowledge during systems design has been a largely informal process, often based on guidelines such as those by Billings for "human-centered" design [1].

In the language of DO-178B/ED-12B (the recommendations for certification of airborne software) such informal processes constitute *reviews*, which are distinguished from *analyses*: "analyses provide repeatable evidence of correctness and reviews provide a qualitative assessment of correctness" [14 Section 6.3]. We are interested in the possibilities of developing analysis methods to augment reviews in HF assessments of new and ongoing designs.

Some researchers in the HF community have taken a path that points in this direction. They develop models: models of automation, of pilots, and of interfaces using techniques from the fields of system and cognitive modeling. Since models are predictive by nature, they can be used to support some forms of analysis at the earliest stages of the automation design process. Examples of this recent trend include [3,8,13,20,23].

A description of automation behavior is a prerequisite for these modelling approaches. The HF researchers concerned have therefore built their own descriptions of autopilot behavior: Degani with the OFM formalism [3], Sherry et al. with Operational Procedures Tables (OPT) [20], Vakil and Hansman with "hybrid" models combining control block diagrams with mode transition matrices [23], and Javaux [6] with diagrams of mode transition conditions. These descriptions are very similar to the formalisms used by computer scientists, and rely on the idea that automated systems can be modeled as finite-state machines.

A branch of computer science known as "Formal Methods" (FM) specializes in modeling the behavior of automated systems using forms of mathematical logic that can be subjected to very powerful analyses using mechanized theorem provers and model checkers. Finite-state machines are among the formalisms used in FM, and researchers in this field have recently started applying their methods to cockpit automation. For example, Butler et al. [2] examine an autopilot design for consistent behavior, Leveson et al. [11] look for constructions that are believed to be particularly error-prone, and Rushby [15] compares an autopilot description against a plausible mental model. Leveson and Palmer [10], and Rushby, Crow, and Palmer [16], show how their methods could be used to predict a known automation surprise in the MD-88 autopilot [12].

This convergence in the modeling approaches used in the HF and FM communities suggests developing new methods that draw on the strengths of both groups: the HF

community provides an understanding of what needs to be modeled and how, while the FM community supplies notations and tools that can subject those models to searching scrutiny. Our purpose in this paper is to suggest how such combined methods support mechanized analyses that can be used during design in a way that will be more thorough, repeatable, and objective than informal reviews.

SUPPORTING THE DESIGN PROCESS

As suggested above, the challenge is to build models, methods, and tools that allow intervention earlier in the automation design process, at the point when high-level behavior of automation is specified. Models can be used to predict the behavior of automation in specific circumstances, to reflect the psychological processes underlying formation of mental models and the way these influence users' behavior of automated systems, and to capture interactions at the human-computer interface (including safety and performance issues).

Addressing each of these topics is a major challenge for the HF and FM communities. By providing answers and solutions, new design techniques can be developed that allow automation designers to test alternative options early in the design process, thereby augmenting traditional automation prototyping techniques. We use two examples—detecting and avoiding undesired scenarios, and investigating incomplete mental models of autopilot behavior—to illustrate application of such techniques to automation design.

DETECTING AND AVOIDING UNDESIRED SCENARIOS

Our first example is based on automatic speed protection in the A320. This protection is invoked to avoid overspeed conditions and causes an automatic mode transition.

Automatic speed protection on the A320

V/S FPA is a vertical mode that allows the pilot to command a vertical trajectory with a specific vertical speed (V/S sub-mode) or flight-path angle (FPA sub-mode). Target vertical speed is specified by the pilot on the FCU (Flight Control Unit) and ranges from -6000ft/min to +6000ft/min (-9.9° to +9.9°). The autothrust, if activated, is automatically engaged in SPEED (or MACH) mode when V/S FPA engages. SPEED controls engine thrust to maintain airspeed to a second target value also selected on the FCU.

V/S FPA and SPEED work together to hold their respective targets (vertical speed and airspeed); however, priority is given to V/S FPA. For example, when a descent is required, V/S FPA commands a pitch attitude that achieves the target vertical speed. SPEED reduces engine thrust, possibly to IDLE, to avoid airspeed increasing (the aircraft is descending) and departing from the target airspeed. However, even IDLE thrust may not suffice if the commanded descent is steep (e.g., 6000 ft/min) and, as a result, airspeed may start to increase beyond the value selected on the FCU. V/S FPA, however, will maintain the same pitch attitude because priority is given to vertical speed over airspeed.

To avoid airspeed reaching dangerous values (e.g., buffet speed), the A320 autopilot features an automatic speed protection that substitutes an OPEN mode for V/S FPA if airspeed reaches maximum acceptable speed for the current aircraft configuration (VMAX or Vfe). OPEN modes are climb or descent modes that have no specific vertical speed target and give priority to airspeed.

FCU selected target altitude plays a very important role in determining which OPEN mode is substituted for V/S FPA: OP DES (open descent) engages if the FCU selected target altitude is below current altitude, otherwise OP CLB (open climb) engages (i.e., if FCU selected target altitude is above current altitude). Activation of the automatic speed protection in descent means that OP DES will normally replace V/S FPA, and immediately decrease pitch attitude to reduce airspeed to the target specified on the FCU.

The protection scheme works very well in this situation and solves problems that may result from engaging V/S FPA with a high rate of descent. There is, however, a scenario where the automatic speed protection, while still achieving its goal of protecting airspeed, leads to an aircraft behavior that deviates dramatically from the pilot's intention.

An Automation Surprise

The unexpected interaction occurs when the aircraft is descending in approach with V/S FPA engaged; for example, if air traffic control (ATC) has required the aircraft to delay the descent process and level-off during final approach. When ATC allows the pilot to resume the descent, the aircraft is located above the glideslope, and has to descend steeply to reintercept the normal descent path. Airbus recommends using V/S FPA in this situation and setting the FCU selected altitude to the missed approach or go-around altitude (in case a go-around were to be performed). The action is undertaken when the aircraft is already below the missed approach altitude, and the FCU selected altitude is therefore set to a target above the aircraft—a very rare situation in normal operations.

Problems with automatic speed protection will appear here if the pilots pay insufficient attention to airspeed and deploy the flaps too early: when the flaps are deployed, maximum acceptable speed is automatically reduced to Vfe, the placard speed for the extended flaps setting. An overspeed situation will therefore occur if the flaps are deployed *before* airspeed has dropped below Vfe.

A reversion to an OPEN mode will occur here, but since the target FCU altitude has been set to the missed approach altitude, which is above the aircraft, the autopilot will not revert to OP DES but to OP CLB (fig. 1). The

pilots are faced with a double automation surprise: a mode reversion occurs, and the aircraft starts climbing when they expect it to descend.

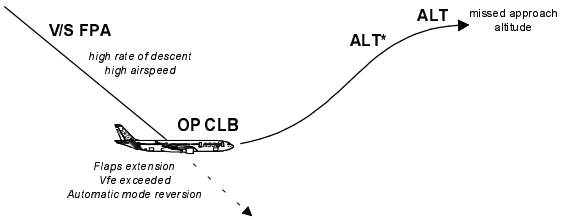

Figure 1. The reversion from V/S FPA to OP CLB

This incident occurred several times in line operations. Airbus therefore decided to offer their customers a 'global speed protection' package in which the reversion to an OPEN mode has been eliminated. Instead, the autopilot remains in V/S FPA and the target vertical speed is adjusted to maintain airspeed below maximum acceptable speed. The same design is installed by default on the A330 and A340.

HF considerations suggest why such incidents occur. Automation surprises can be related to incomplete mental models of automation behavior [11]; the automatic speed protection is rarely, if ever, activated in a pilot's lifetime, and frequential effects are likely to be at play here [7]. Moreover, anticipating the reversion is a complex cognitive operation [6]: the pilot has to recall a series of steps (i.e., his approximate mental model of the transition), articulate them with the prediction that flaps deployment will lead to an overspeed condition, and assess the fact that V/S FPA is engaged and the FCU selected altitude is set—unusually—above the aircraft altitude. This is very unlikely for a mundane action such as deploying the flaps.

We would like to be able to predict the potential for this surprise during design of the autopilot, and to be able to explore alternative designs that might mitigate it. One approach would be to run simulations of the design, but that raises the problems of choosing the scenarios to run, and the behaviors to look out for. A formal methods technique known as *model checking* is able to examine all possible scenarios, and to check them against desired invariants or mental models.

Mechanized Analysis of the Design

The mechanized analysis presented here was undertaken in the Murphi state exploration system [4] from Stanford University, using verification techniques described in detail in [15]. We summarize that approach here, but omit the details of our analysis due to space constraints.

The basic idea is to construct a state-machine model of the relevant aspects of the A320 automation and then explore all possible scenarios for that model. Simple expectations for consistent behavior can be checked by evaluating an *invariant* at each step, while more complicated expectations can be examined by checking the state of the automation against that of a mental model, which is also represented as a state machine. State exploration, a variant of model checking, is able to explore all possible scenarios because the models are abstracted so that only a finite number of states needs to be considered. In this example, we do not need to model the exact speed of the aircraft, nor the values of its protection limits: all that matters is whether the aircraft is above or below the relevant limit, and these relationships can all be represented by just a few values.

Systems are modeled in Murphi by specifying their state variables and a series of rules describing the actions the system can perform and the circumstances under which it performs them. Properties that should hold in some or all states are specified as assertions or invariants, respectively. Murphi then performs an exhaustive simulation of the system, examining all possible behaviors and verifying that the specified properties are indeed satisfied in all reachable states. If a property fails to hold, Murphi generates an error trace that describes a scenario leading to the violation.

At the level of abstraction germane to our analysis, the behavior of the autopilot can be described in terms of six state variables representing the vertical mode, FCU selected altitude, max speed constraint, aircraft speed, flight phase (e.g., descent), and flap configuration. With the exception of the flap configuration, which has been further abstracted to a Boolean variable indicating whether or not the flaps are extended, these variables range over a set of uninterpreted constant values. For example, the variable representing the current maximum-allowable aircraft speed may take one of two values: VMAX or Vfe, representing the maximum-allowable speeds for V/S FPA and "flaps extended" modes, respectively. This simple example is encoded in approximately ten Murphi rules, including the "Startstate" rule, used to specify the initial state of the system. In our model, the initial state corresponds to the aircraft configuration in normal descent: vertical mode is V/S FPA, FCU altitude is below aircraft altitude, max speed is VMAX, aircraft speed is below VMAX, flight phase is descent, and the flaps are clean. Other Murphi rules correspond to engaging V/S FPA mode, engaging OPEN mode, setting the flaps, entering the GO AROUND altitude in the FCU, increasing, decreasing, or maintaining aircraft speed, and so forth.

Given even this very simple Murphi model representing partial mode logic for an autopilot, we can explore design options for the overspeed protection mechanism. Let us assume we are designing such a mechanism and want to analyze the behavior of an overspeed protector that automatically transitions the autopilot from V/S FPA mode to an OPEN mode (to achieve FCU selected altitude

independently of FMGS-entered altitude constraints) if the aircraft speed exceeds the current maximum allowable speed, which in our model would be either VMAX or Vfe. To explore this design option, we need only add a Murphi rule corresponding to the constraint for the overspeed condition, and an invariant that asserts basic expectations about autopilot behavior. For example, we might specify that if the aircraft is descending, the autopilot will never transition to OP CLB mode. If we now allow Murphi to perform state exploration (which it does by firing the specified rules in all possible sequences), we discover an unanticipated and potentially dangerous coupling between the overspeed protection mechanism and the mode logic for OPEN modes: if the aircraft is descending with flaps set and its altitude is below the FCU selected altitude, then OP CLB, rather than OP DES is selected.

This behavior (which corresponds to the automation surprise described in the previous section) seems undesirable, so we consider a different overspeed protection mechanism. Instead of transitioning to an OPEN mode if aircraft speed exceeds MAX-allowable speed in V/S FPA, we remain in V/S FPA mode, but reduce the target vertical speed commanded by the pilot. If we modify the Murphi overspeed condition rule accordingly and repeat the state exploration, no problems are detected, arguing strongly for the merits of the second design.

Of course, this example is an after-the-fact reconstruction of known events in the design history of the A320, but it illustrates how finite-state verification can be used to explore design options and to inform design choices during the early stages of development.

The Murphi model of the vertical mode logic described thus far specifies only the system model and simple expectations expressed as an invariant. If we add an explicit representation of the pilot's mental model of the mode logic (again using techniques described in [15]), we can explore deeper aspects of the autopilot design relative to pilot interaction. For example, if we assume, following [6,7], that a pilot's accumulated experience and training induce simplified mental models in which rarely-encountered transitions are omitted, then we can compare a proposed new or retrofitted design with that of a pilot's simplified mental model to see if these are prone to diverge from each other in a potentially unsafe way.

Returning to our example, the occurrence of the automatic transition from V/S FPA to OP CLB mode during descent on approach is sufficiently unusual that it is plausible to assume a generic mental model in which this transition is not represented. Predictably, when we validate this simplified mental model against the Murphi system model for the first overspeed protection mechanism, Murphi reports an error trace that corresponds to the automation surprise reflected in the incident scenario; the V/S FPA mode predicted by the generic pilot model is at variance with the OP CLB mode calculated by the vertical mode logic.

INCOMPLETE MENTAL MODELS

The notion of mental—or conceptual—model is central to many recent papers on pilot-automation interaction [7,15,20,21,23]. Mental models influence users' behavior, and therefore have an impact on safety. Since they are usually simplified and incomplete representations of system behavior (cf. [7,15]), an important issue is to determine how far they can be simplified without endangering safety and efficient operations. We refer to the simplest such models as *minimal safe mental models*.

Minimal Safe Mental Models

Javaux is presently conducting a study for Airbus Industrie to determine the minimal mental model required for operating the A340-200/300 autopilot safely and proficiently [8].

A reverse-engineered description of autopilot behavior has been used to write a questionnaire that investigates this question. The questionnaire has been submitted to seven Airbus instructors and test pilots. The experts were asked to rate the importance given to each of the conditions involved in the 268 possible mode transition scenarios on the A340-200/300 (i.e., 'how safe is the pilot-automation system if the pilot doesn't know that this condition must be fulfilled for the transition to occur?'). A typical result is shown in figure 2. It describes the conditions for dual autopilot engagement (using Javaux's diagrams, cf. [6]). The numbers below the conditions correspond to the average rating given by the experts (1 means 'not at all important,' and 4 means 'very important').

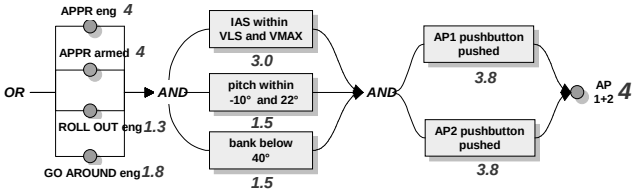

Figure 2. Rated diagram for dual AP engagement

By defining a threshold of 2.5 (the middle value between 1 and 4), the following minimal safe model for dual autopilot engagement emerges (figure 3).

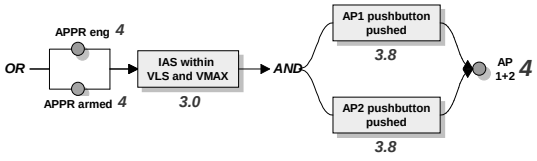

Figure 3. Minimal safe mental model for dual AP engagement.

How reliable are the results obtained by this type of subjective evaluation method? When asked about their

strategies for rating the conditions, the experts explained they were looking for situations or scenarios where ignoring the conditions threatened safety. They used their own experience and their recollections of incidents and accidents to uncover typical difficult cases. While heuristic, the approach is not systematic and exhaustive, and is likely to miss some interesting cases as it did in the OP CLB automation surprise scenario. The results obtained to date in the Airbus study show indeed that experts' ratings are convergent on some mode transitions, but differ widely on others. These results clearly underscore the limitations of subjective evaluation techniques.

Analysis of Safe Minimal Mental Models

FM techniques and tools provide a way to assess and quantify the variability of models derived via subjective evaluation. Using the finite-state verification techniques applied previously, we can validate whether or not a given model is both minimal and safe relative to an autopilot design. There are several possible strategies for applying finite-state verification techniques to the problem of identifying minimal mental models relative to a design. Given that we want efficiently to explore a range of psychological models, we propose to encode Javaux' s rule rating scheme directly in the Murphi model of pilot behavior, thereby allowing us to parameterize the selection of model for a given run of the verifier, and iteratively to test models of increasing (or decreasing) complexity, corresponding to lower (or higher) rule condition ratings. Comparison of the rating-augmented model against the model of the actual system design or retrofit via finite-state verification allows us to identify the minimal model of pilot behavior that is consistent with the actual system design, thereby confirming (or denying) the empirically-derived minimal model, as well as predicting potential automation surprises at precisely those points at which the pilot and system models diverge. Armed with this information, system analysts and designers have the opportunity to make informed decisions about where to apply "human-centered" design principles to bring the pilot and system mode logic models into alignment, and to anticipate where additional pilot cues or training may be necessary.

The notion of a minimal safe model is necessarily relative to a model of a given aircraft. Nevertheless, once the safety of a particular minimal mental model has been validated, we can also run that model against a design for a next-generation aircraft, and predict areas of convergence and divergence relevant to design decisions and pilot training.

DISCUSSION

Previous work in Formal Methods has lacked psychologically interesting models, while work in Human Factors has lacked automated, replicable methods and tools. The approach presented here addresses both of these deficiencies: this paper shows the importance of having psychologically interesting mental models for automated exploration of design space, and, conversely, of having fully automated, replicable methods for analyzing these models and using them to calculate and predict potential automation surprises in the design specifications of aeronautic systems.

There is much excellent work in the interdisciplinary field of Human-Computer Interaction (HCI) that seeks to understand and reduce the sources of operator error in automated systems. The combined approach described here, which extends previous work in Human Factors and in Formal Methods, is intended to complement and enhance, but not replace, ongoing work in HCI. Our approach uses existing finite-state verification methods and tools to validate empirically-derived psychological models of pilot behavior, and to calculate the consequences of these models for designs of pilot-autopilot interaction. The novelty of the approach lies in the fact that we combine methods and tools from both the human factors and formal methods communities, and use these techniques to analyze automatically properties of system design, whereas most previous work is grounded in one or the other of these disciplines and uses manual techniques to analyze artifacts of system deployment (including automation surprises). We view automation as a valuable adjunct to, but certainly not a replacement for, thoughtful human review of design specifications. The exhaustive search of the design space and the informative error trace provided by finite-state verifiers are assets that can be easily assimilated into existing manual review processes.

In the future we plan to apply our combined approach to larger examples and to evaluate its effectiveness in more realistic design applications, possibly including Javaux's models of the A340-200/300. We are also interested in using the technique to probe further the consequences of incomplete or inappropriate mental models, including the interaction between a mental model of normative behavior and a system model with one or more anomalous modes; to examine the interactions between multiple mental models (e.g., a model for each crew member); and to anticipate and assess guidelines for training materials and checklists.

ACKNOWLEDGMENTS

The work of Denis Javaux was supported by Airbus Industrie. The work of Judith Crow and John Rushby was

supported by SRI International and by DARPA through Air Force Rome Lab contract F30602-96-C-0204.

REFERENCES

1. Charles Billings. *Aviation Automation. The Search for a Human-Centered Approach*. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.
2. Ricky Butler, Steven Miller, James Potts, and Victor Carreño. A formal methods approach to the analysis of mode confusion. In *17th AIAA/IEEE Digital Avionics Systems Conference*, Bellevue, WA, October 1998.
3. Asaf Degani. *Modeling Human-Machine Systems: On Modes, Error, and Patterns of Interaction*. Ph. D. thesis, Georgia Institute of Technology, 1996.
4. David Dill. The Murphi verification system. In Rajeev Alur and Thomas Henzinger, editors, Computer-Aided Verification, CAV '96, volume 1102 of Lecture Notes in Computer Science, pages 390--393, New Brunswick, NJ, July/August 1996.
5. Denis Javaux and Véronique De Keyser, editors. *Proceedings of the 3rd Workshop on Human Error, Safety, and System Development (HESSD'99)*, University of Liege, Belgium, June 1999.
6. Denis Javaux. The cognitive complexity of pilot mode interaction: A possible explanation of Sarter & Woods classical results. In *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero'98), Montréal, Canada*, May 1
7. Denis Javaux. A method for predicting errors when interacting with finite state machines. In Javaux and De Keyser [5].
8. Denis Javaux and Estelle Olivier. Assessing and understanding pilots' knowledge of mode transitions on the A340-200/300. In *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero'00)*, Toulouse, France, September 2000.
9. Richard Jensen and Lori Rakovan, editors. *Proceedings of the Eighth International Symposium on Aviation Psychology*, Columbus, OH, April 1995.
10. Nancy Leveson and Everett Palmer. Designing automation to reduce operator errors. In *Proceedings of the IEEE Systems, Man, and Cybernetics Conference*, October 1997.
11. Nancy Leveson, L. Denise Pinnel, Sean David Sandys, Shuichi Koga, and Jon Damon Rees. Analyzing software specifications for mode confusion potential. In C. W. Johnson, editor, *Proceedings of a Workshop on Human Error and System Development*, pages 132–146, Glasgow, Scotland, March 1997.
12. Everett Palmer. "Oops, it didn't arm." - A case study of two automation surprises. In Jensen and Rakovan [9], pages 227–232.
13. Peter Polson, Sharon Irving, and J. E. Irving. Applications of formal models of human computer interaction to training and use of the control and display unit. Final report, System Technology Division, ARD 200, Federal Aviation Administration, Dept. of Transportation, 1994.
14. Requirements and Technical Concepts for Aviation, Washington, DC. *DO-178B: Software Considerations in Airborne Systems and Equipment Certification*, December 1992. This document is known as EUROCAE ED-12B in Europe.
15. John Rushby. Using model checking to help discover mode confusions and other automation surprises. In Javaux and De Keyser [5].
16. John Rushby, Judith Crow, and Everett Palmer. An automated method to detect potential mode confusions. In *Proceedings of 18th AIAA/IEEE Digital Avionics Systems Conference*, St Louis, MO, October 1999.
17. Nadine Sarter and David Woods. Pilot interaction with cockpit automation I: Operational experiences with the flight management system. *International Journal of Aviation Psychology*, 2(4): 303-321, 1992.
18. Nadine Sarter and David Woods. Pilot interaction with cockpit automation II: An experimental study of pilots' mental model and awareness of the flight management and guidance system. *International Journal of Aviation Psychology*, 4 (1): 1-28, 1994.
19. Nadine Sarter and David Woods. "Strong, Silent, and 'Out-of-the-loop' ": Properties of advanced (cockpit) automation and their impact on human-automation interaction. Technical Report CSEL 95-TR-01, Cognitive Systems Laboratory, The Ohio State University, Columbus, OH, February 1995.
20. Lance Sherry, Peter Polson, Michael Feary, and Everett Palmer. Analysis of the behavior of a modern autopilot. Technical Report C69-5370-009, Honeywell Incorporated, Minneapolis, MN, 1997.
21. Lance Sherry and Peter Polson. Shared models of flight management systems vertical guidance. *International Journal of Aviation Psychology*, 9(2):139-153, 1999.
22. Sanjay Vakil, R. John Hansman, Alan Midkiff and Thomas Vanek. Feedback mechanisms to improve mode awareness in advanced autoflight systems. In Jensen and Rakovan [9], pages 243-248.
23. Sanjay Vakil and R. John Hansman. Approaches to mitigating complexity-driven issues in commercial autoflight systems. In Javaux and De Keyser [5].