APPLYING A COMMON CONCEPTUAL MODEL APPROACH TO VERTICAL NAVIGATION AUTOMATION

Sanjay S. Vakil and R. John Hansman

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

1. Introduction

The increased levels of automation in current generation aircraft have lead to a new set of human factors related concerns. This set of problems is suspect in a large number of incidents. These include a crash caused by autopilot disconnect confusion in an A300 at Nagoya[1], incorrect mode selection in an A320[2] at Strausbourg, overspeed problems with the B757[2] and several others. Billings[3] has synopses of many of the incidents in the last two decades. These problems have raised the efforts of the design and human factors communities and, more recently, were the focus of the FAA Task Force Report on Interfaces Between Flightcrews and Modern Flight Deck Systems[4].

Modern flight deck systems have evolved, incrementally, from earlier designs within the operational constraints of the National Airspace System. The evolution has consisted of the incorporation of additional functionality intended to provide more flexibility to flight path design and to improve safety through the inclusion of automated warning systems. In some aircraft these warning systems have been given sufficient authority to override pilot control in hazardous situations.

One of the primary goals of these highly automated systems was to reduce the number of flight incidents attributed to flight crew error. In actuality, the proportion of errors attributed to the flight crew has remained both relatively constant and the dominant cause of aircraft incidents[4]. The form that these errors have taken has changed over time, and one important area is in confusion between what pilots expect the system to do and what the Autoflight System (AFS) actually does in operation. These types of problems have been referred to as “Mode Awareness Problems”.

Several solutions have been put forth to deal with these concerns. The solutions vary in both the times necessary for their implementation and in the levels of comprehensiveness. Some problems that appear in aircraft automation appear to be of an “introductory” nature and appear during the early operational life of an aircraft. As an example, the Airbus A320 was introduced in 1984 and was evaluated in 1992 as having about 3 hull losses per million departures. When re-evaluated in 1995, hull losses were reduced to about 1.25 per million departures. While not attempting to draw statistical significance from the “rare events” captured in these numbers, it appears that the critical period in new aircraft automation is during the early years of its usage. The “shakedown” period that occurs early in the operational lifetime is the most critical.

To deal with these introduction problems, near term, and relatively inexpensive, solutions are often implemented. These may consist of procedural and training modifications. Procedures are regularly updated with changes designed to work around problems with existing automation. On a larger scale, there is work underway on an advanced Vertical Navigation (VNAV) trainer[5]. This tool is designed to show pilots the underlying complexity of the VNAV system and the implications of a particular set of mode choices.

Mid-term solutions consist of enhancing feedback in the cockpit to prevent confusion and to allow pilots the ability to accurately predict what automation will do next. This will involve the installation of new displays, or at the least a modification of existing display software, causing this solution to be more expensive and longer term. After the Strasbourg accident, displays which better differentiated between vertical descent modes were available as retrofit options on the A320 and were standard equipment on the newer A319. More comprehensive solutions include an Immediate Mode Management Interface[6] Hutchins has presented for advanced cockpits and an Electronic Vertical Situation Display[7] shown in Figure 1.

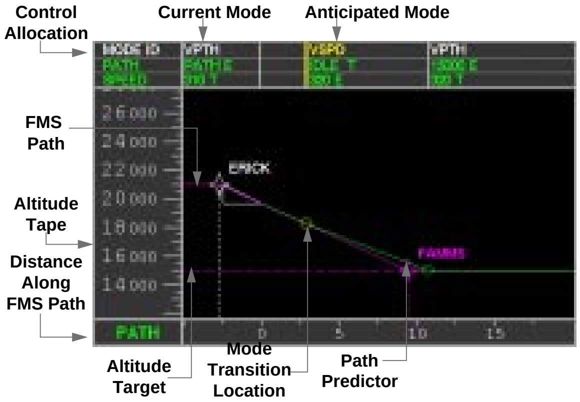

Figure 1: Electronic Vertical Situation Display

The ultimate solution is to modify existing automation and to anticipate the next generation of new aircraft automation to create systems which do not suffer from Mode Awareness Problems, or are at least created with an understanding of the relevant issues. The cost of such an overhaul (of not only the automation, but also the training) is likely to only be undertaken when new functionality in the cockpit requires redesign.

Option	**Cost/Time**
Procedure Changes	Low
Training Changes	Low-Medium
Improved Feedback	High
Automation Modification	Very High

Table 1: Mode Awareness Problems Mitigation Options

In order to support the development and certification of new complex automation systems which consider flight crew operational understanding, a Common Conceptual Model (CCM) approach is proposed. A CCM is designed to be a high level functional representation of automation designed to cater to the capabilities and limitations of the operators. For systems developed in an operator-directed manner, such as aircraft automation, the CCM will ultimately become the functional design specification for testing and certification purposes, and the basis of training material.

2. Absence of Conceptual Models in Current Automation Systems

A review was conducted of the existing flight automation systems as part of a study of mode awareness problems in current aircraft. There does not appear to exist a simple, consistent global model of flight automation articulated to certification officials or pilots. This result is based on the available training literature, 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 studies: B757, B767, B747-500, A320, A300, MD-11 and F-100.

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. 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 simplified rule-based, 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.

Compared to other automation systems, clear mental models of time-critical flight systems are of particular importance. In current aircraft automation, the pilot is normally given final control and full responsibility. This implies that the pilot must understand, at some level, all automation behaviour in order to intervene effectively and appropriately in emergency situations. It may be the case that a limiting factor on aircraft automation design should be the level of complexity that an operator can maintain and readily access as a mental model.

The goal of this work is not to attempt to directly manipulate the mental models of the pilots, but rather to provide an accurate and complete representation upon which to base individual models. The cognitive science and training communities can then use these representations as the basis of training regimes and material.

3. Common Conceptual Models

A CCM is a high level functional representation of the automation which is common to, and consistent between, pilots, certification officials and design engineers. A notional diagram of a CCM is shown in Figure 2. Operators are expected to be the most important group in defining the CCM, since they interact with the automation in high tempo situations without ready access to detailed underlying knowledge of the system. As such, the CCM is intended to be a representation of the functionality of the automation at a level which is accessible and useful to the operator. Ideally, a CCM will be introduced in the early stages of design and maintained throughout the lifecycle of the automation and be the basis of the system design framework. The ultimate goal of having a CCM is to reduce the automation problems during the operation of the system by having the system better understood by its users.

A CCM is especially useful in systems where a human operator has final control authority and responsibility. For these systems, the CCM will enable an operator-driven methodology to allow the limitations, needs and capabilities of the operator to be addressed early in and throughout the design cycle. In addition, the CCM can be used as a limiting factor on the complexity of aircraft automation design.

Emergency and non-nominal operations are a key issue in the definition of a CCM representation of the automation. Non-nominal situations are a problem because of both their rarity and the fact that they may be atypical. Since mental models are created during nominal operations (where the vast majority of operator experience occurs), they may not hold (or may even be a liability) in emergency situations. It is critical that these situations must be both reinforced in training.

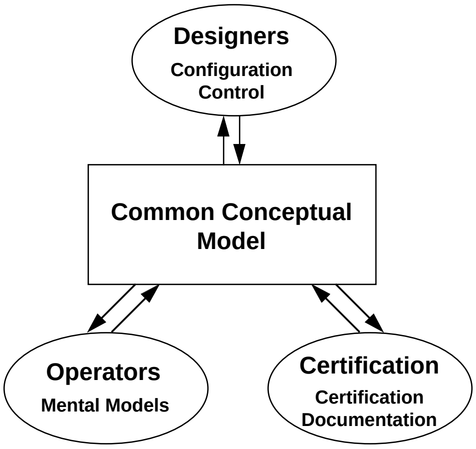

Figure 2: Notional Diagram of Common Conceptual Model

Underlying the concept of a CCM is the premise that having some sort of consistent representation is an advantage over the disconnected representations which appear to be the current state in aircraft automation. Currently designers, operators and certification officials involved with aircraft automation use different system models to support their tasks, and have differing mental representations of the system. It is important to note here that a CCM is not intended to fully specify the entire models used by each of these groups. Instead, the CCM is intended to be a functionally complete basis to use as the source material for the construction of these models. The requirements that must be derived for complete system design will necessarily be more complete in order to implement the system details. The form of the representation ultimately used by these groups may also be significantly different; as an example, designers may need to create pseudocode. The CCM is more accurately described as a functional abstraction of the automation defining a contract which must be fulfilled. The CCM then could be used for the design of AFS functionality and software, the development of training material, the certification of AFS operation and ongoing configuration management.

4. User Abstraction Level

The CCM is intended to be a representation of the functionality of the automation at a level which is accessible and useful to the operator. Since the level at which this representation is presented is critical to system implementation and certification, during early stages of design, it will be necessary to determine the abstraction level at which to create the CCM. This is an operator-driven issue: the determination of abstraction level is dependent on the skills, training and aptitude of the intended audience. It would be useful if a CCM could be used to explore the levels at which the automation should be functionally specified.

This problem can be considered by the level of goals intended to be solved independently by the automation. In a very low level representation, the goals are limited and short term and only able to deal with a low level of uncertainty. The operator must translate their much higher level goal into the language of the lower level goals. By contrast, higher level abstractions handle the decomposition of high level goals into low level goals independent of the pilot.

It appears that the tradeoff is between flexibility, or functionality, of the automation and the level of specification detail which an operator must maintain:

the complexity of the automation. To obtain a system which is simple to use, the automation must make a set of assumptions about how the high level functions which are selected by the operator will be performed. The choices associated with these assumptions are moved out of the hands of the operator, thereby making the system simpler, but reducing flexibility. However, if the operator is required to explicitly specify all of the assumptions, the system complexity can quickly become overwhelming. An operator-directed CCM design may help determine the level at which functionality should be presented to the operators.

At one end of the abstraction spectrum, the CCM could be created at an extremely "low" level, where the details of each mode change are made explicit, concerning transition details, duration, timing, gains, etc. At this level of abstraction very little is left to implementation details, but the detailed specification is unlikely to be useful to the operator, as internalizing this information will be difficult. However, giving the operator access to the underlying details may afford a higher level of flexibility. Since the operator has to construct a flight plan from these very low level elements, the details can be optimized by the operator.

At the other extreme, a CCM could be created which specifies very "high" level functions for the automation. Riley[8] has created a representation of functionality at the Air Traffic Control directive level. Essentially, directives from ATC can be programmed into the flight automation directly where they are parsed and acted upon. This enables the operator to interact with the system in a specified, "natural language" mode and avoids the explicitly detailed specification of the preceding example. However, in order to complete these high level functions, the automation designers must make assumptions about how these functions are best completed.

As an example, consider the manners in which an aircraft can gain altitude, or "Flight Level". If the maneuver is critical, altitude could be gained at the expense of airspeed: the aircraft's speed will be allowed to decrease so it may climb as quickly as possible. However, if the gain in altitude is less critical, it may be more prudent to maintain airspeed during the climb. Both of these manners ("not speed protected" and "speed protected") have instances in which they are applicable, but a high level of abstraction may place the choice in the hands of the automation. This may lead to a loss of flexibility. It can be argued that the language and syntax of the specification can be enriched to allow the specification of "how", but this creates another set of operational assumptions which the operator must internalize. This can lead (in an extreme case) back to the low level specification described earlier.

4.1 Suggested Representation of a CCM

The representation used for a CCM must be able to support multiple levels of abstractions. To this end, a hierarchical model is likely to be an applicable choice. To further organize the system, hierarchical levels of the representation should be distinct -- any interactions between levels completely defined and independent of higher levels. An example of a hierarchically structured representation is shown in Figure 3.

Several techniques currently exist which may be used to represent a hierarchical system. Some of these support direct hierarchical usage, such as the SpecTRM model[9], the OFAN model[10], and many of the other extensions to state transition diagrams. Control diagrams and block diagrams are used by the controls and human factors communities to represent a variety of complex interactive phenomena. Another class of representation is rule-based models, where abstractions can be made by grouping rules and their arguments into larger logical units. A final class is the linguistic set which creates hierarchies by the creation of a language and syntax which can be combined in meaningful

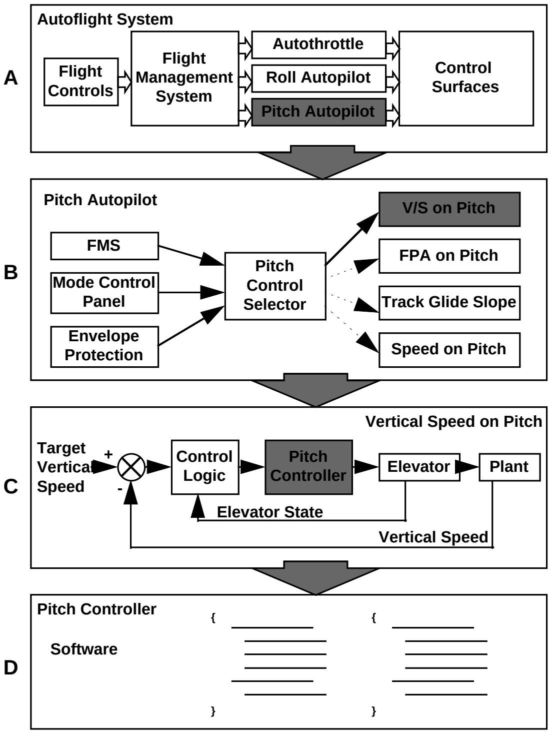

Figure 3: Hierarchical Structure

manners.

To date, these approaches have only been applied in accounting of the current, existing system, rather than an organizational tool for the design of new systems. Further, the manners in which these techniques account for the current system do not convert readily into training material. It is hoped that a technique and process can be designed which will enable both system organization and a more transparent conversion to training material.

4.2 Determination of User Abstraction Level

During design and early evaluation, the abstraction level at which the CCM is specified may remain in flux. However, in an operator-directed methodology, the abstraction level needs to be determined by the limitations, tasking level and capabilities of the anticipated operators. A system which has an invariant operating regime may be able to be abstracted at a very high level by allowing assumptions to be made by designers. A more dynamic system may require abstraction at a low level in order to provide the flexibility to deal with a changing operating environment. The flexibility required by the operation environment must be used as the basis for abstraction level specification.

How to determine the level of abstraction is a matter of significant import on which work is currently underway. The issues and tradeoffs in this decision require careful analysis. From a conceptual standpoint, the goal is to engage the operator at a level which keeps them actively involved in the system but which does not overwhelm their cognitive capabilities.

As an example of this type of evaluation, consider transitions between different operating modes in an autoflight system. This example is particularly applicable because previous work[6] has shown that a substantial number of problems occur during automatic or uncommanded mode transitions in current system. Consider examining the “Syntactical” Complexity of a particular transition[11]. The idea behind this measure is to examine the Boolean operations which must be evaluated to determine if a mode transition is to occur. This sort of evaluation is the essence of how computers work.

By examining the number of independent criteria which must be evaluated, it has been suggested[11] that we can gain insight into the complexity of the operation. Taking this to the next step in an operator-directed approach, this may be correlated with the short-term memory limitations under which humans operate, of being able to maintain 5-9 elements[12]. If the number of Boolean choices exceeds this number, it may be necessary to abstract the system at a higher level. By using a higher level of abstraction and “chunking” together independent criteria by making assumptions of the operation of the system, the cognitive load may be reduced. This evaluation is related to GOMS modeling but attempts to do the evaluation in early system design rather than after it has been designed.

The example below demonstrates the criteria which can switch an aircraft autoflight system into a vertical speed mode. Either the operator can press the V/S (Vertical Speed) button, the operator can switch the ALT (Altitude) target while in the ALT ACQ (Altitude Acquisition) mode, or the GS (Glide Slope) can be lost while in the GS mode. In this example, only two variables need be maintained in short term memory at any time: enough to evaluate each of the independent OR statements.

Transition to Vertical Speed Mode:

If

V/S button pressed *OR*

ALT changed *AND* in ALT ACQ mode *OR*

GS lost *AND* in GS mode

Figure 4: Example of Transition to Vertical Speed

Another way in which a higher level of abstraction can be enforced is through having the system operate in a manner analogous to another with which the operators are familiar. Current work[8] takes this approach by “recasting” the autoflight system into the language of ATC directives. This approach can be very effective (the “Desktop” analogy in some modern computer Graphical User Interfaces), but requires that the consonance between the designed system and the analogous representation be accurate and sufficiently complete. It may be the case that using an analogous representation inappropriately may cause dissonance where the two models disagree. Similar results[13] regarding the consonance of alerting criteria and operator compliance have been found. Pilots in closely spaced approaches were found to comply with alerts more often when the alerting criteria were in consonance with their own mental models of these criteria.

How to evaluate the “reduction in complexity” of using analogies is still an open matter which must be resolved.

5. Applying Common Conceptual Models to VNAV Systems

It appears that aircraft Vertical Navigation systems are an excellent candidate to use as an example of the concept of Common Conceptual Models. These systems are part of highly complex automation systems with the capability of providing additional functionality to pilots and airlines. Unfortunately, the current implementations have poorly documented behaviour and have been implicated in a number of flight incidents and accidents. This has led to some airlines training pilots on the VNAV systems, and others explicitly specifying that they should not be used.

The CCM approach suggested in this paper acknowledges, and designs around both the nominal and non-nominal flight regimes. This is particularly challenging. From a design standpoint the limiting factor in VNAV system complexity appears to be that pilots need to retain understanding of the system at all stages of flight in order to effectively intervene in abnormal situations.

This often leads to the design constraint on the system being the complexity inherent in the non-nominal operations rather than in the normal operations.

Using a CCM may provide a basis for critiquing designs early in the creation process and evaluating these designs for their impact on operator’s mental facilities. Other approaches which have been suggested are currently focusing on nominal operations. Ultimately these approaches, which may be applicable during nominal operations, will need to be extended to include non-nominal and emergency situations. Extending these systems may prove difficult.

Three ideas are presented here to contrast the possible approaches to a VNAV system design. Each of these approaches corresponds with techniques that could be used in a CCM framework to reduce the complexity of the existing VNAV system. The ideas being presented are a very simple VNAV system and a goal hierarchy approach.

The initial step in a CCM design is to create a set of functional requirements which the automation must satisfy. This set of requirements is then analyzed to determine how they may be fulfilled.

5.1 Simplified VNAV System

One approach to creating a simple CCM is to create a very simple set of functional requirements. In this example, the functions required of the automation are to attain and maintain a vertical speed, or path, target and an airspeed target. This simplified VNAV system is created by the removal of the idea of individual independent modes from the automation. Instead, we use a continuous control space paradigm. A VNAV path consists of a set of linked trajectories, each defined by a path target and a speed target. The path target is a trajectory joining two points and is defined by its altitude or altitude variance. The speed target is a simply the current airspeed of the aircraft. The target values could be entered a variety of ways, but all directly control the path and speed parameters.

This is a very low level abstraction of a VNAV system since it requires the operator to reduce long term high level goals into a set of shorter term goals which can be entered into the VNAV interface. By creating such a small set of possible behaviours for the system, complexity is reduced at the expense of additional overhead in actually using the system.

	Pitch Control	Thrust Control
Path	X	
Air Speed		X

Figure 5: Control Allocation Matrix for Simplified VNAV System

This system can also be simplified by using only the pitch to control the vertical speed and only the thrust to control the airspeed, as shown in Figure 5. The control allocation between the control mechanism and the control variable remains constant, and so the control allocation matrix is only partially populated.

By creating a reduced set of requirements, we have a very simple and elegant VNAV system which is consistent with a simple CCM containing no ambiguities. The system resembles early generation autopilots. However, the operator is responsible for any problems that occur, and for dealing with envelope protection violations. This system can be characterized as having high predictability and low capability.

5.2 Goal Hierarchy System

A more functional system can be created by creating a more complex set of interacting and competing goals which the automation is attempting to accomplish. By doing so, functionality can be added to the system to allow flight planning, envelope protection and “speed protected” capability. However, such a system gains functionality at the cost of complexity. In contrast to the simplified system, Figure 6 shows that all four states of the control allocation matrix are multiply populated in a manner consistent with current autoflight systems. This leads to a more complex automation system with which the operator must interact.

	Pitch Control	Thrust Control
Path	-Vertical Speed -Altitude Capture -VNAV-Path -Glide Slope	-Flight Level Change -Envelope Protection -VNAV-Spd
Air Speed	-Flight Level Change -Envelope Protection -VNAV-Spd	-Vertical Speed -Altitude Capture -VNAV-Path -Glide Slope

Figure 6: Control Allocation Matrix for a more Complex VNAV System

In order to manage this level of complexity, some organization needs to be imposed on the system. Goal hierarchy systems create an explicit hierarchy of the various goals the automation is attempting to accomplish. By articulating these goals in an understandable manner, the operation of the system can be determined by the interactions between these goals and their relative importance (locations in the hierarchy). Two possible hierarchies is shown in Table 4. Note that final authority can be placed in either the automation via envelope protection, or in the pilot via manual control.

Goal Hierarchy A

1. Safety: Envelope Protection
2. Manual Goal: follow manual pilot inputs
3. State Limit: limit set by pilot
4. Target Goal: attain/maintain specified target
5. Trajectory Goal: follow programmed trajectory

Goal Hierarchy B

1. Manual Goal: follow manual pilot inputs
2. Safety: Envelope Protection
3. State Limit: limit set by pilot
4. Target Goal: attain/maintain specified target
5. Trajectory Goal: follow programmed trajectory

Table 2: Example Goal Hierarchies (in decreasing importance)

In contrast to the simplified system, a goal hierarchy can give rise to a much richer set of interactions to assist in attaining higher level goals. For example, a Flight Level Change (flight to a specified altitude) can be created by setting an altitude State Limit (flight level) and a thrust Target Goal (CLIMB). When the aircraft reaches the specified flight level, the level off will arise from an interaction between the State Limit and Target Goal. This approach is similar to a Subsumption Architecture[14]. However, the interaction of each of these differing goals must be clearly defined to the operators. A serious concern is that the emergent behaviours of these interactions may become excessively difficult to manage, even though the underlying rules are straightforward.

6. Conclusion

It appears that an operator-directed Common Conceptual Model methodology could be a useful approach to the design of complex systems requiring supervisory control by human operators. This paper has discussed some of the concepts surrounding the creation of such a model and how it might be used. Having a representation which can be viewed at multiple levels of abstraction may allow the selection of an appropriate level of abstraction, in consonance with human cognitive capabilities. The VNAV system is an interesting test case for the use of this methodology as it is a complex system which is undergoing problems with its current implementation. Several conceptual models of the VNAV system will be cursorily explored in the context of a Common Conceptual Model.

The work on linking human limitation to level of abstraction is still very young and it is hoped that this paper may offer a starting point for the discussion. In addition, the structure for the representation for a CCM is still under research.

7. Acknowledgments

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. Mecham, M (1994). Autopilot Go-Around Key to CAL Crash *Aviation Week & Space Technology*. May 9, 1994.

2. Hughes, D. (1995). Accidents Direct Focus on Cockpit Automation, *Aviation Week & Space Technology*. January 30, 1995.
3. Billings, Charles (1997). Aviation Automation: The Search for a Human-Centered Approach. Mahwah, NJ: Laurence, Erlbaum Associates
4. 1996 FAA Task Force Report on the Interfaces Between Flightcrews and Modern Flight Deck Systems.
5. Chappell, A. R. & Mitchell, C. M. (1997). Intelligent tutoring systems for ’trained novices’: Building expertise in mode management. Proceedings of the 1997 International Symposium on Aviation Psychology. Columbus, OH.
6. Hutchins, E. (1996). Immediate Mode Management Interface Final Report. NASA Ames Research Center.
7. Vakil, S. (1996). Development and Evaluation of an Electronic Vertical Situation Display. MIT Aeronautical Systems Laboratory Report ASL-96-2
8. Riley, V. A. (1997). The Cockpit Control Language: A Pilot-Centered Avionics Interface.
9. Leveson, N.G., Palmer, E. (1997). Designing Automation to Reduce Operator Errors. Proceedings of Systems, Man, and Cybernetics Conference.
10. Degani, A. (1994). Modelling Human-Machine Errors: on Modes, Error and Patterns of Interaction. Doctoral Thesis. Atlanta, GA: Georgia Institute of Technology.
11. Javaux, D. and De Keyser, V. (1998). The Cognitive Complexity of Pilot-Mode Interaction. A Possible Explanation of Sarter and Woods’s classical results. In G. Boy & C. Graeber (eds), Proceedings of the International Conference on Human-Computer Interaction in Aeronautics, May 27-29, Montreal, Canada.
12. Barsalou, L. W. (1992). Cognitive Psychology, An Overview for Cognitive Scientists. Mahwah, NJ: Laurence, Erlbaum Associates.
13. Pritchett, A. (1997). Pilot Non-Conformance to Alerting System Commands During Closely Spaced Parallel Approaches, MIT Aeronautical Systems Laboratory Report, ASL-97-2.
14. Brooks, R. (1986). A robust layered control system for a mobile robot. IEEE J. Robotics and Automation 2:75.