Explaining Sarter & Woods' Classical Results.

The Cognitive Complexity of Pilot-Autopilot Interaction on the Boeing 737-EFIS.

Denis Javaux (1)
Work Psychology Dpt
University of Liege, Belgium
d.javaux@ulg.ac.be

1. This research was carried out thanks to a contract agreement with SFACT/DGAC (Direction Générale de l'Aviation Civile) in France, within the framework of a study group on human factors. The author would like to thank the people at the SABENA's flight training department where the research is performed, and in particular Franco Scala and Jean de Clercq, for their support, their technical help and their kindness.

Summary:

This paper attempts to explain – and predict – the kind of results obtained by Sarter and Woods and other authors regarding the interaction with the automation on modern highly computerized aircraft. The paper suggests that the cognitive complexity of the sub-tasks involved in the interaction with the autopilot's modes is actually increased by some choices made by the manufacturers. This hypothesis is tested on two sub-tasks related to commanded and uncommanded mode transitions. A metrical measure of complexity and a method for predicting the pilots mental models, the knowledge gaps and the associated errors are proposed and evaluated. The predictions are in strong accordance with the observations in operational settings. The results of this research are of potential use for the certification, design and training processes.

1. Introduction

Pilot-automation problems or difficulties have been these recent years among the 'hot' topics of discussion in the human factors community in aviation. Several authors contributed to that, in particular Wiener (1989), Sarter & Woods (1992, 1994, 1995) and Billings (1997). Many experimental researches and operational feedback have shown that besides the obvious benefits of automation for commercial aviation there were some aspects of pilot-automation interaction that were unsatisfying from the pilot's perspective. Sarter and Woods' researches (1992, 1994, 1995) in particular have had a deep impact, because they showed very clearly that highly qualified and proficient pilots had obvious lacks in their understanding of automation. The difficulties were situated both a the declarative (erroneous or incomplete mental models) and procedural (limits in the ability to modify automation status and behavior) levels. A recurrent statement about automation in the literature is that it is 'too complex'.

Complexity however is never defined, at least in ways that could lead to its operationalization for improving the automation as it exists now. This paper addresses this question. It attempts to provide new means to measure complexity, in particular when applied to pilot-automation interactions. A measure of complexity and a method for predicting knowledge gaps and errors are proposed and applied on the automation of the Boeing B737-EFIS. This aircraft is the one studied by Sarter and Woods in 1992.

The objective of this research is clearly to understand what happens when pilots interact with the automation and how human performance can be predicted when its structural and functional properties have been analyzed. Similar goals are pursued by other researchers (e.g. Degani, 1996; Leveson & Palmer, 1997). The goal is to produce models that allow to explain (at least a subset of) what is already known (e.g. some of Sarter & Woods' results) and to predict how some modifications applied to present or future automated systems can affect human performance. Such models would allow cockpit manufacturers to compare design alternatives and to reject the options that lead to degraded human performance (e.g. more confusion errors). They would also help the people involved in the certification process to evaluate, compare and accept or reject the design options considered by the manufacturers.

We believe that such models can be built, by gathering contributions from cognitive and experimental psychology, cognitive models in HCI, artificial intelligence and complexity theory: modern cognitive science is now more integrated and mature than in the past and it can really impact the way we design man-machine systems (see Wickens, 92 for a good example of how cognitive and experimental psychology can be applied to understanding and improving man-machine interaction situations).

2. Computational and Cognitive Complexity

To gain a better understanding of what complexity is, and of how it can be grounded on theoretical and experimental cognitive psychology, we will focus on a few examples related to the modes of the B737-EFIS. The modes are the basic functions of the autopilot (for good and general discussions on modes, see Degani & al, 1998 and Degani, 1996). Each mode is capable to execute a specific task (e.g. to maintain a specific heading or to climb to a new altitude). The first two examples have been described by Sarter & Woods in 1992. They concern the 'gulf of execution' (fig. 1) on the automation (as it has been named by Don Norman in 1988), that is the ability to control the modes engagement status. The third example is drawn from our own observations in the B737-EFIS cockpit and some interviews with the pilots. It exemplifies Norman's gulf of evaluation (fig. 1), that is the ability to infer the mode engagement status from the information explicitly presented to the pilot.

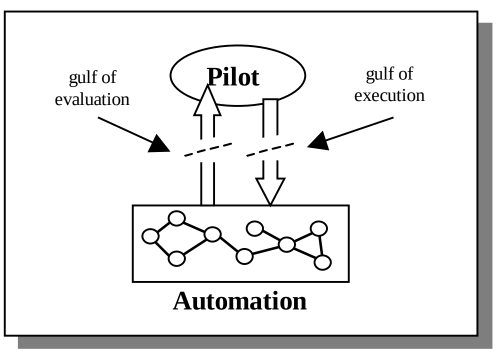

fig. 1. the gulf of evaluation and the gulf of execution (Norman, 1988)

2.1. The engagement of ALT HOLD

In usual conditions, the preconditions and actions for engaging the ALT HOLD mode on the B737-EFIS are rather simple: whatever the pitch (vertical) mode currently engaged, the pilot simply has to push the ALT HOLD switch on the MCP panel to engage the mode (if the aircraft is not already flying at the target MCP selected altitude). This is simple and straightforward. During the approach phase however, with the APP mode engaged (i.e. LOC and G/S engaged), pressing the ALT HOLD switch will not work! The autopilot is in a protected state, mostly because the aircraft is close to the ground. It is not possible to substitute an alternate mode (e.g. ALT HOLD) to LOC and G/S without first disconnecting the autopilot. The sequence of actions to perform is therefore more complex: the autopilot must be first disconnected, either by pressing one of the two TOGA switches (and this is not always compatible with the current context or with the pilots goals), by detuning the ILS frequency, or by pressing one of the two autopilot disconnect switches. At this stage, the APP mode is still engaged if any of the flight directors is still on. Both FDs must be turned off to fully disengage APP (i.e. LOC and G/S). The autopilot can now be re-engaged in its command state (CMD) and the ALT HOLD mode finally engaged by pressing the ALT HOLD switch on the MCP panel.

The preconditions and actions for engaging a mode (e.g. ALT HOLD) are therefore dependent on the context (here the APP engagement state). In the example above, if APP is not engaged, the sequence of actions is short, 'natural' and easy to execute. If it is engaged, the sequence is long, awkward and difficult to perform.

2.2. Thrust control during aborted take-offs

Sarter and Woods (1992) describe the problems that arise when pilots are requested to execute an aborted take-off at low speeds on the B737-300 (one of the three aircraft in the B737-EFIS category). Pilots are usually trained in the simulator for aborted take-offs at high speeds (ground speeds close to the V1 threshold speed).

Aborted take-offs at speeds close to V1 represent the most difficult and hazardous situations during the take-off phase, because the aircraft must be stopped as soon as possible (before the end of the runway is reached). At these high ground speeds (in fact, at any speed greater than 64 kts), the automatic system (the autothrottle or A/THR) that manages the thrust of the engines is in the THR HOLD mode: the thrust settings are constant and stabilized at their N1 nominal take-off value (TO/RTO). In case of aborted take-off, the action to take is to immediately set the thrust levers at their idle position. This has two effects: an automatic disengagement of the autothrottle system (i.e. the pilot regain a full manual control of the thrust settings) and a reduction of the engine thrust, which contributes to slow the pace of the aircraft. This quick response is a part of the normal procedure in case of aborted take-off. The action for getting into the loop and regaining manual control is simple, 'natural' and straightforward.

The behavior of the autothrottle system differs when aborted take-offs are attempted at speeds inferior to 64 kts. Below 64 kts, the A/THR is in a thrust mode identified as N1. The function of N1 is to increase the thrust of the engines to the N1 TO/RTO nominal target value set before take-off. The thrust levers move accordingly. In case of an aborted take-off attempted when N1 is still engaged (i.e. below 64 kts), the A/THR will not disengage and will remain in N1 when the pilot takes the action to retard the thrust levers to idle! The thrust levers will moreover automatically move forward if they are released by the pilot because the A/THR and the mode N1 are still engaged. The pilot and the automation are in conflict. To prevent this, the correct procedure for aborted take-offs below 64 kts is therefore to first disengage the A/THR – and its N1 mode – by a specific action on the MCP panel and then to reduce the thrust to idle.

Problems arise here because the actions to execute for reaching the goal (transforming the decision to stop the aircraft into specific actions on the man-machine interface) depends on the context (i.e. the current ground speed) in which the decision is made: two actions are necessary for ground speeds inferior to 64 kts when only one is needed (reducing the thrust to idle) for the higher speeds. There is therefore no simple mapping in the example 2.2. (and in the example 2.1) from the pilot's intention to the correct sequence of actions to execute. The gulf of execution is widened when it should actually be narrowed. See Norman (1988) or Wickens (1992) for examples of good mappings and how they allow to reduce (cognitive) complexity.

2.3. The interpretation of Flight Mode Annunciations

Flight mode annunciations appear in the B737-EFIS on the Flight Mode Annunciator (FMA), which is situated above the EADI (Electronic Attitude Director Indicator). The annunciation of the modes on top of the EADI or PFD (the Primary Flight Display in Airbus terminology) is common to most of the manufacturers (Boeing & McDonnel Douglas, Airbus, Bombardier).

Flight modes annunciations allow the pilots to infer what is the current status of the modes, in the pitch, roll and thrust channels. A mode can be either engaged (active), armed (ready for engagement) or disengaged (inactive). The interpretation of flight mode annunciations is a kind of translation process: the symbols presented on the FMA are 'translated' into a mental representation of the current modes engagement status. Problems of interpretation can occur on the B737-EFIS when the pitch modes are engaged.

Pitch (or vertical) modes are particular on most aircraft because they are in general asymmetrically coupled with the thrust modes: when a pitch mode is engaged, a specific thrust mode is automatically engaged. Pitch and thrust modes are coupled because both relate to the (potential and kinetic) energy of the aircraft. When ALT HOLD is engaged on the B737-EFIS, the AFDS (Autopilot Flight Director System) automatically engages the MCP SPD thrust mode, whose role is to control the speed of the aircraft. Both modes are active at the same time. The coupling of these modes allows the aircraft to fly level at a constant speed specified by the pilot. Because of the coupling, the pilot can - and has to - process two annunciations in order to infer which pitch mode is currently engaged: MCP SPD - ALT HOLD will mean that the ALT HOLD mode engaged, FMC SPD - VNAV PTH will mean that the VNAV (vertical navigation) mode is engaged,…

Things become more complex with the LVL CHG (level change) mode. When LVL CHG is used for a climb, the annunciation is N1 - MCP SPD. There is no explicit mention on the FMA of a level change. When used for a descent, the annunciation is either RETARD - MCP SPD or ARM - MCP SPD, depending on the position of the thrust levers. Here again, no mention of a level change. The LVL CHG acronym is never displayed as for the other pitch modes (such as ALT HOLD, ALT ACQ, V/S, VNAV and G/S). The cognitive processes the pilot has to install to infer that LVL CHG is engaged are obviously (slightly) more complex than for the other vertical modes.

One would like to see LVL CHG as an exception. Things would be fine if the engagement status for the other pitch modes was clearly indicated on the FMA. This is not always the case: in some situations, the pilot is faced with the FMC SPD - ALT HOLD annunciation. One is driven to believe that the autopilot is here in ALT HOLD. In fact, the current vertical mode is VNAV! The normal annunciations for VNAV are either FMC SPD - VNAV PTH or FMC SPD - VNAV SPD, but if a MCP selected altitude is interposed in a climb or a descent toward a FMC target altitude, the aircraft will automatically level at the intermediate altitude and the FMA annunciation will change from FMC SPD - VNAV PTH or FMC SPD - VNAV SPD to FMC SPD - ALT HOLD. VNAV remains engaged here but this not explicitly stated on the FMA and the pilot is really *driven* to think that ALT HOLD has been substituted to VNAV. The cue to use to detect that VNAV is engaged is the thrust channel annunciation: FMC SPD is always annunciated for thrust (the thrust is managed by the FMC) when VNAV is engaged. Relying on the pitch mode annunciation - while very useful when VNAV is explicitly displayed - is misleading here.

Similar oddities occur with the approach (APP) mode. When selected, APP invokes the arming and the engagement of LOC and G/S. These modes are annunciated in their respective roll and pitch columns on the FMA by VOR/LOC (either white or green) and G/S (either white or green). In the last seconds of the approach (at 27 feet RA exactly), the annunciation changes from MCP SPD - G/S - VOR/LOC to MCP SPD - FLARE - VOR/LOC, because the FLARE mode has been automatically substituted to G/S. The APP mode has been armed or engaged during the whole process, but its acronym – while being used on the MCP panel for selecting the mode – has never been shown on the FMA.

This example with the APP mode demonstrates how poorly the man-machine interface reflects the hierarchical relationships that exist between the modes. The autopilot modes can be thought off as acting at different functional levels: APP is explicitly composed of LOC, G/S and FLARE. Lateral (or roll) modes such as HDG SEL or VOR/LOC on the other hand contains sub-modes that are not annunciated. The respective role of these sub-modes is to capture and hold a target heading. The capture and holding behaviors are here implicit, hidden in submodes that are never annunciated as such. For the vertical (or pitch) modes however the sub-modes for the transition to, the capture and the holding of a target altitude are explicit, with the dissociation between LVL CHG or V/S, ALT ACQ and ALT HOLD being explicitly presented to the pilot. The information provided on the interface is therefore inconsistent and incomplete. It does not reflect the hierarchical relationships between the modes and the sub-modes. This makes the elaboration of a correct mental model of the mode hierarchy and behavior more difficult.

These examples show again how the inference process - the translation from the annunciations to a representation of the current mode status - is made more complex by rules that do not straightforwardly map one space of information (the flight mode annunciations) onto the other one (the flight mode engagement status). The rules are complex and present exceptions. The gulf of evaluation is widened when it should be narrowed.

2.4. Cognitive complexity

We have seen how the gulfs of evaluation and execution are actually widened because the functional properties of the automation and its interfaces render cognitive processes context-dependent, and therefore more complex and prone to error. *Evaluating* the current engagement status of the modes (example 2.3) and *generating* and *executing* actions for modifying this status (examples 2.1 & 2.2) is made more complex by the absence of a simple mapping between two worlds, the mental representations of the pilots (interpretations and intentions) and their counterparts in the pilot-automation interface (indicators, displays, controls and their functional properties). This is typical of many high-tech systems (Norman, 1988), and in particular of modal systems (Degani, 1996).

Complexity here directly stems from a gulf or a gap, a gap that the designer must attempt to narrow. As suggested in the examples 2.1, 2.2 and 2.3., the size of the gap can be measured by the complexity of the function that translates the information on one side (e.g. an intention) into the information on the other side (e.g. a sequence of actions to perform on the interfaces). The size of the gap is a measure of the computational complexity of the cognitive processes the human operator must execute. This is the idea behind *cognitive complexity*, a concept which has been applied successfully in other domains (Kieras & Polson, 1985).

In the case of the aborted take-off with the A/THR engaged, cognitive complexity is actually increased because the pilot has to execute a conditional test before performing the actions on the man-machine interface. The number of mental operations to perform for achieving the task is increased (i.e. perceiving the actual speed of the aircraft, checking if it is inferior to 64 kts), extra information must be maintained in working memory (i.e. the actual speed of the aircraft) and in long-term memory (i.e. the sequences of actions for each possible outcome of the conditional test) and as a consequence additional attentional resources must be allocated to the achievement of the task for reaching error-free performance. The following definition of cognitive complexity can be proposed:

> *the cognitive complexity of a situation (where some tasks are performed) is measured by the quantity of cognitive resources that must be allocated to achieve the tasks with a performance level greater than a given threshold.*

The cognitive resources mentioned in the definition are the number of operations to perform, the amount of information to maintain in working or long-term memory and the amount of attentional resources to allocate for performing the tasks (Javaux, 1997; Javaux & De Keyser, 1997). Reducing complexity in a given man-machine situation usually means reducing any or all of these quantities. The performance level can be measured by different means (e.g. the rate of errors, the time to perform the tasks,...) and as a general rule the higher the performance threshold the greater the cognitive complexity.

This approach to cognitive complexity is derived from the notion of *computational complexity* used in theoretical computer science (Alliot & Schiex, 1993). We feel it offers a good starting point for addressing the kind of problems described by Sarter & Woods as well as those described in the examples. One of the interesting things with the definition is that cognitive complexity is inherent to the achievement of any task: any situation is complex because there are cognitive processes at play - with their specific complexity - in any situation.

The examples above demonstrate how the cognitive processes involved in a given situation - and their complexity - are influenced by the structural and functional properties of the automation and its interfaces. Is it possible to produce well-defined methods for deriving cognitive complexity from the specifications of the automation and its interfaces? Despite the difficulties that arise from the analysis of cognitive processes that resort to covert behaviors (see Javaux, 1997 or Javaux & De Keyser, 1997 for a discussion on this point), we believe that such methods can be built and used for improving the automation on the flight deck.

3. Pilot-mode interactions

Pilot-mode interactions are the most visible part of pilot-automation interactions. The automation on the flight deck however should not be solely reduced to the modes: they are many automated systems in the cockpit that do not resort to the modes: the autopilot trim, the integrated displays such as the EADI and EHSI,….

To study the cognitive complexity of the interaction with the modes, it is necessary to understand pilot-modes interactions in a larger context than the mere consideration of commanded and uncommanded transitions and mode awareness. The task of interacting with an autopilot mode can be decomposed into seven sub-tasks shown on the figure 2: (1) predicting the behavior of the aircraft when the mode is engaged, (2) predicting the behavior of the pitch, roll and thrust channels when the mode is engaged, (3) recalling or generating a correct sequence of actions for setting the mode's parameters, (4) recalling or generating a correct sequence of actions for modifying the mode's engagement state, (5) interpreting the flight mode annunciations and the status of the MCP switch lights, (6) interpreting the behavior of the aircraft and the behavior of the pitch, roll, and thrust channels, and finally (7) predicting uncommanded modes transitions.

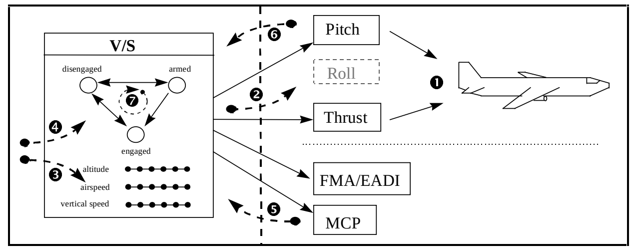

fig. 2. the sub-tasks of the pilot-mode interaction

The dissociation into simpler interaction sub-tasks can be used to classify the different types of pilot-automation difficulties or problems encountered in operational settings: e.g. (1) difficulties for predicting the behavior of VNAV when engaged (Sarter & Woods, 1992), automation surprises (Sarter & Woods, 1994, 1995), (2) difficulties for predicting - and assessing - the side-effects of the autopilot on the stabilizer trim (e.g. Moscou, 1991; Nagoya, 1992; Orly, 1994), (3) problems encountered when programming the CDU or MCDU (Sarter & Woods, 1992), (4) difficulties for disengaging the APP mode after G/S engagement or reducing the thrust in case of aborted-takeoff at low speeds (Sarter & Woods, 1992), (5 & 6) difficulties for evaluating the current vertical mode engaged, as in Mont St-Odile's accident (Monnier, 1993), and (7) difficulties for predicting uncommanded or 'indirect' mode transitions (Sarter & Woods, 1992, 1995), automation surprises (Palmer, 1995).

A *thesis* of this paper is that each of these examples can be tackled as a problem of cognitive complexity. Most - if not all - of the sub-tasks above can be studied as 'translation' tasks, that is as (cognitive) functions that map two spaces of information. Their computational or cognitive complexity can therefore be studied and evaluated. Each function has to be seen as a gulf or a gap that must be narrowed. It is therefore necessary to understand how the structural and functional properties of the automation contribute to the cognitive complexity of the interaction. This is mandatory for identifying the potential improvements that will reduce the cognitive complexity of the pilot-mode interactions and therefore make the man-machine system safer and more reliable.

Several methods dedicated to the investigation of the cognitive complexity of the sub-tasks 4 (recall or generation of actions sequences for commanded mode transitions) and 7 (prediction of uncommanded mode transitions) have been developed. They are presented in the sections 5 and 6 of this paper.

4. Commanded and uncommanded mode transitions (sub-tasks 4 & 7)

The sub-tasks 4 and 7 have been chosen for validating the cognitive complexity approach. Both have been shown to introduce possible weaknesses in operational contexts (Sarter & Woods, 1992, 1995). Other tasks such as the task 1 (predicting the behavior of the aircraft when a mode is engaged) and the task 5 (inferring the mode engagement status from the explicit information in the cockpit) will be studied in a later phase.

To derive the cognitive complexity of the sub-tasks from the description of the functional and structural properties of the automation, a first step is to analyze the characteristics of the automation that affect their achievement. For the sub-tasks 4 and 7, this means describing the rules that govern the mode transition behaviors: when do modes become armed, engaged or disengaged?

4.1 The description of the transition rules

Describing the different rules that govern the mode transitions has been more difficult than initially thought. There is no single source where the information needed to describe the rules can be found. The Operations Manual for the B737-EFIS has been used during the initial phase and a *content analysis* of the chapter devoted to automatic flight (chapter 7) has been performed. Each page of the manual has been divided into sub-sections and each sub-section has been extensively documented. A content analysis was mandatory because the information for a single mode is generally distributed between several sections of the Operations Manual. The conditions for the uncommanded disengagement of V/S for example are described in the sections about the performance and minimum speed reversions (07.20.17, 2 & 3), not in the sections explicitly dedicated to V/S (07.10.03, 5; 07.10.05, 3 & 07.20.04, 4).

The results of the content analysis have been used to describe the different transitions for each mode of the autopilot. It rapidly proved unsatisfying: the Operations Manual is sometimes incomplete, incoherent or erroneous. Several functional properties of the modes known to the instructors are not described in the Operations Manual (e.g. the annunciation FMC SPD - ALT HOLD for VNAV when intercepting an intermediate altitude). The Maintenance Training Manual for the aircraft, several structured interviews sessions with instructors and a maintenance engineer as well as several simulators sessions were therefore needed to complete the model.

The following data structure, with five data fields, has been used for describing the rules. The examples concern the rule devVS2 (commanded engagement of V/S).

- an unique *identifier*: it describes the type of the transition (e.g. an engagement,…), its commanded or uncommanded character and the mode that it concerns. A numerical index is added to the description if there is still a possible confusion with an other rule. The identifier devVS$_2$ means that the rule describes a commanded (v) transition of the V/S mode (VS) from the disengaged (d) to engaged state (e). It is the second (2) rule with these characteristics for V/S.

- a *natural language description*: it describes in natural language when and how the transition occurs.

  V/S is engaged when the ALT ACQ mode is engaged and a new MCP altitude is selected by rotating the altitude selector. The amplitude of the altitude change must be greater than 100 ft.

- some *quotes* from the Operations Manual when they are available. They support the natural language description of the rule. They help to locate the source of the information, especially when it is distributed in several sections of the Operations Manual.

  V/S is (automatically) activated (engaged) by pressing the V/S mode switch or when a new altitude is selected when ALT ACQ is annunciated (07.10.02B, 5). The V/S mode automatically engages when the ALT ACQ mode is engaged and a new altitude is selected which is more than 100 feet different than the previously selected altitude (07.20.05,1)

- a *Boolean description* of the rule: the rule is described in the language of propositional logic. The different conditions for the rule to be triggered are determined and connected by means of the classical AND, OR and NOT Boolean operators.

(ALT ACQ engaged) AND (MCP altitude changed) AND (amplitude of change > 100)

- a *procedural description* of the transition rule. This is a very important part of the characterization. The procedural description adds the temporal dimension to the Boolean description of the rule. For the transition to occur, the conditions must usually be verified in a specific order, with a sequential, partial or concurrent ordering of the conditions. Taking into account the temporal dimension is essential for understanding and predicting some types of errors. The graphical form associated with the procedural description also allows to enrich the representation, by means of colors and shapes. A rule is composed of three parts (see fig. 3): the *preconditions* ('ALT ACQ engaged') are the conditions that have to be met before the rule can be applied. They define the context in which the rule can be used. The preconditions are surrounded by a dotted rectangle. The *conditions* ('MCP altitude changed' and 'amplitude of altitude change > 100') are the conditions that have to be met for the transition to be triggered. Preconditions and conditions are displayed in colored boxes. The colors code for the type of the condition (i.e. pilot-dependent or pilot-independent conditions, conditions related or not to the automation). A colored circle finally represents the *terminal state* of the transition. The color of the circle (green, white or black) depicts the type of the transition (respectively an engagement, arming or disengagement). The acronym of the mode whose engagement status is changed after the transition (V/S in the figure) is associated with the final node. Seeing the rules in their procedural form is of great help for the researcher, as well as for the instructor and the trainee. They can therefore be used for training.

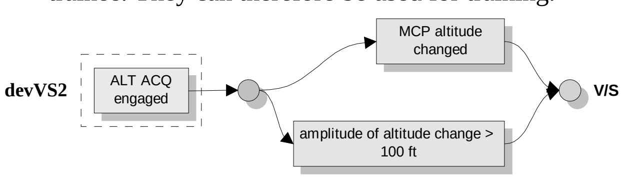

fig. 3. the procedural description of the rule $devVS_2$.

This scheme of description has been applied to the B737-EFIS. Four modes (ALT HOLD, ALT ACQ, LVL CHG and V/S) have been analyzed and characterized with the five dimensions. Forty-four rules have been elicited. They cover the arming transitions {disengaged}⇨{armed}, the engagements {disengaged, armed}⇨{engaged} and the disengagements {armed, engaged}⇨{disengaged} for the four modes on the B737-EFIS in Sabena configuration (the detail of the rules may differ, depending on the configuration of the systems).

5. Syntactical complexity

A measure of complexity has been developed. The measure of *syntactical complexity* is a *metrical* measure of complexity. It is applied to the *Boolean* representation of the rules to produce a numerical value. It attempts to evaluate the mean number of cognitive operations the pilot has to perform to apply the rule in the operational context. This measure is considered as *local*, because it applies to the rules taken in isolation, without considering the possible interactions between the rules (e.g. the similarities that facilitate the storage and the maintenance of the rules in long-term memory). A more *global* – and powerful – approach is presented in the section 6.

.

(ALT ACQ engaged) AND (MCP altitude changed) AND (amplitude of change > 100)

fig. 4. the logical expression for the rule $devVS_2$ (V/S engagement)

The approach considers that to use a rule such as the one ($devVS_2$) presented in the figure 4, the pilot has to evaluate if the different arguments (e.g. ALT ACQ engaged, MCP altitude changed,…) are true or not. This is needed to decide whether the transition will occur or not, if necessary by performing the actions for the transition to be triggered (e.g. change the MCP altitude). The mean number of operations to perform for evaluating the truth value of the Boolean expression is our first measure of complexity. It is named the 'syntactical complexity' because it depends on the syntactical structure of the expression, a hierarchical tree obtained by analyzing the relationships between the Boolean operators NOT, AND, and OR in the rule.

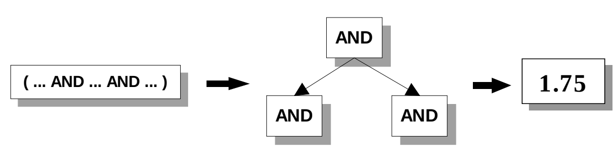

fig. 5. the syntactical complexity of the rule $devVS_2$

According to this method, the syntactical complexity of the rule $evVS_2$ is 1.75 (fig. 5): it is necessary to carry out in average 1.75 evaluations of arguments (e.g. is ALT ACQ engaged) before being able to decide if the transition is going to be triggered or not in the current context (defined by the truth values of the respective arguments in the expression). A mathematical method based on a series of recursive functions has been developed to compute the syntactical complexity associated with these Boolean expressions. C-Plex, a computer software has been developed to apply the evaluation method to a whole set of rules. It has been used to compute the syntactical complexity associated with the ALT HOLD, ALT ACQ, LVL CHG and V/S modes and their forty-four transitions.

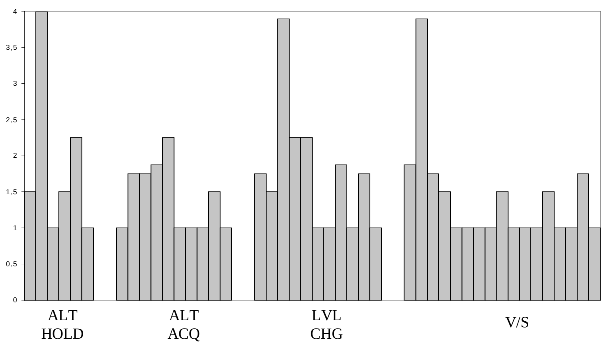

fig. 6a. syntactical complexity

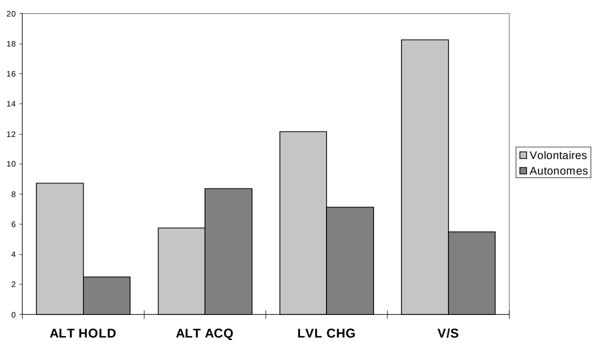

fig. 6b. additive syntactical complexity

The syntactical complexity for the individual rules, grouped by modes are presented in the figure 6a. Three rules in particular stand out. They correspond to the commanded transitions for engaging ALT HOLD, LVL CHG and V/S, with the complex conditions and actions to verify in case of APP engagement (see above the example 2.1). These three measures therefore correlate with the results already obtained by Sarter & Woods in 1992 (the explanation lies in the rule chaining mechanism, see the section 6.3 below).

The additive measure for each mode is obtained by adding the values computed for each individual transition rule. They are presented in the figure 6b. The graph dissociates commanded (light gray) from uncommanded (dark Grey) mode transitions. The mean number of operations to perform while using the different modes is higher in case of commanded transitions than in case of uncommanded transitions. ALT ACQ is an exception. This is a mode whose transitions are mostly uncommanded. V/S on the other hand has a greater (additive) syntactical complexity than the other modes, in part because it has three engagement states (disengaged, armed and engaged), and therefore more transitions possibilities.

6. The explanation and the prediction of the errors and knowledge gaps

The ultimate goal of the cognitive complexity approach is to relate the kind of problems or difficulties experienced by the pilots to the functional properties of the automation (i.e. its specific transition rules). As noted above, the researches of Nadine Sarter and Dave Woods (1992, 1994, 1995) have demonstrated that – sometimes wide – gaps exist in the knowledge the pilots have of the transitions and of their triggering conditions: the mental models are incomplete and sometimes erroneous.

We have identified many of these *knowledge gaps* and *related errors* on the B737-EFIS, thanks to the original descriptions of Sarter and Woods, to multiple discussions with the Sabena instructors and pilots, and to several informal observation sessions in the simulator. It appears that some of the forty-four mode transitions are clearly unknown to some pilots (e.g. the mode reversions) or that the representation of the rules and their conditions is sometimes partial (some of the specific conditions are not known) or erroneous (false belief). Explaining these knowledge gaps and the errors they induce is the objective of this section of the paper.

We must first notice the inability of the syntactical measure of complexity to achieve this goal. This measure quantifies the amount of computations needed for evaluating the truth value of a transition rule. Its contribution to a general theory of mode error production is therefore very limited. One must resort to other approaches to account for the problems experienced by the pilots.

Four such *mechanisms* or contributing *factors* are proposed hereafter. They draw upon ideas introduced in Javaux (1997) and (Javaux & De Keyser, 1997). The first two mechanisms (frequential simplification and inferential simplification) show how the transition rules are actually stored in a simplified or prototypical form that is responsible for most of the knowledge gaps observed in the pilot population. The combination of the two mechanisms allows to predict how the pilots mental model will be shaped for a given set of transition rules. The third mechanism deals with the planning of the actions in the cockpit. The last one concerns the interactions with the man-machine interface.

6.1. Frequential simplification

A first hypothesis is that a major contributing factor to the errors and knowledge gaps observed with the pilots is the *frequency* of the pilots exposure to the transition rules. The pilots experience the transition rules and their conditions with different frequencies, either in their normal operational life or in recurrent training sessions in the simulator. Some rules are rarely triggered (e.g. most of the transition rules that concern the GA mode are seldom used in normal operational conditions) while others are used on every flight (e.g. ALT HOLD or VNAV engagement). For a nice study on mode usage, consult Degani & al (1995).

Two measures of frequencies must be distinguished: *rule frequencies* relate to the frequency with which each rule is actually triggered. Some rules are frequently used, others are almost never activated out of the context of the recurrent training sessions (e.g. the mode reversions). *Conditional frequencies* on the other hand concern the frequency with which each individual precondition and condition in a given rule is true when the rule is applied. Rule and conditional frequencies are independent: some rules can exhibit very low frequencies (because they are almost never applied) but contain conditions whose frequency is very high (they are almost always true when the rule is activated).

An accurate evaluation of the different frequencies for the rules and for the conditions is very difficult to obtain. One should ideally resort to a combination of real-flight observations and computerized data logging during extended periods of time. A more suitable method consists of interviewing the instructors. This is the solution we have selected at this early stage of the investigation on frequencies. The approach however is inaccurate because it relies on subjective representations and evaluations. Some of the instructors moreover rarely fly in operational conditions. To circumvent this limitation, a systematic and exhaustive questionnaire will be submitted in the near future to a handful of the pilot-instructors - they are frequent fliers - at Sabena.

The effects of the rule and conditional frequencies are differential. They have different impacts on the rules and their storage. The main effect of the *rule frequency* is to weaken or strengthen the trace of the rule in long-term memory. Rules that are frequently activated have strong traces while rules that are rarely used or triggered have weak traces. The rules whose trace is weak are particularly prone to the influence of the second mechanism (inferential simplification). See the figure 15 for an example. The main effect of the *conditional frequencies* is to lead to a simplification of the rule stored in long-term memory. The rule is stored in a simplified or *prototypical* form. The differences between the prototypical form and its complete and correct representation account for some of the knowledge gaps that have to be explained.

A remarkable example of simplification concerns the *commanded* engagement of the ALT HOLD mode in order to level off at the current barometric altitude. The conditions for engaging ALT HOLD are simple: the G/S mode must not be engaged and the ALT HOLD switch has to be pressed (fig. 7. See also the example 2.1. above).

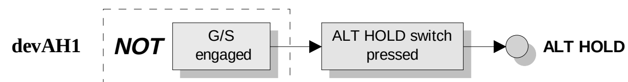

fig. 7. the (complete) representation of the rule $devAH_1$.

The frequency of the rule devAH1 is high (the rule is activated several times during a normal flight). The conditional frequency of the NOT(G/S engaged) condition is particularly high: it is close to one. The condition is usually true - G/S is not engaged - when the pilot attempts to level off at the current altitude (level-offs at the current altitude with G/S engaged can occur in final approach to cope with a request from the ATC but this is not a frequent event). The high conditional frequency of the G/S precondition leads to the following representation of the rule $devAH_1$: pressing the ALT HOLD switch will engage the ALT HOLD mode. The representation of the conditions is altered, the G/S condition is dropped, and the rule is stored in the simplified or prototypical form shown in the figure 8. The effects of the knowledge gap (i.e. the missing G/S condition) can be observed when the pilot attempts to use the rule when G/S is actually engaged. One sees the pilot repeatedly pushing the ALT HOLD switch - without success.

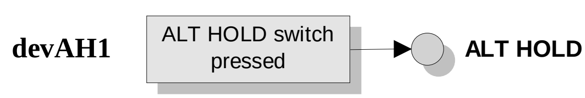

fig. 8. the simplified or prototypical representation of the rule $devAH_1$.

Another nice case concerns the *uncommanded* mode transition from GA to ALT ACQ (with the autopilot engaged in dual channel configuration). The GA (pitch) mode is used to command go-around thrust and a specific pitch or body attitude in case of a go-around. The aircraft climbs towards the final (highest) go-around altitude in the missed approach procedure as a result of GA engagement. GA is then supposed to disengage and be replaced by the ALT ACQ mode ($devAA_2$ transition) when approaching the target altitude. The figure 9 shows however that two specific conditions have to be met before the ALT ACQ mode can be autonomously engaged: the distance to the target altitude must be greater than 1000 ft upon GA engagement and the stabilizer trim position must be compatible with a single autopilot configuration.

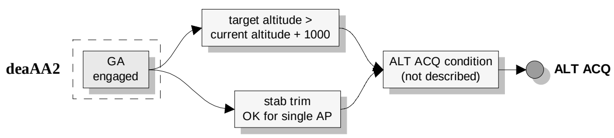

fig. 9. the (complete) representation of the rule $devAA_2$.

It appears that some pilots store a simplified representation of this rule. Frequency effects can once again be invoked as the main explanatory factor. First, the rule itself is seldom used (the rule frequency is low) because go-arounds are rare in normal operational practice. Second, the target altitude and stabilizer trim conditions are true in most of the rare events a go-around has to be performed (their conditional frequencies are close to one): on most airports, the final go-around altitude is sufficiently high for the first condition to be satisfied. And the side-effects of the modes engaged prior to the go-around (e.g. G/S) are usually limited so that the second condition (stab trim) is true most of the time the verification is performed by the automatic system. As a consequence, a simplification is also observed here. A (erroneous) prototypical representation is stored in LTM. It states that the aircraft will climb until the usual ALT ACQ condition (i.e. the one found for the LVL CHG and V/S modes) is achieved. The autopilot will then substitute ALT ACQ to GA (see figure 10). The two conditions have been dropped from the representation. The prototypical representation is inherently hazardous, because in case of over-reliance on the automation, the pilot will not check if the transition to ALT ACQ did really occur when the target altitude was reached. If any of the two conditions is false (e.g. a final target altitude too close to the altitude where GA is engaged), the aircraft will continue to climb and therefore potentially violate the local altitude constraints.

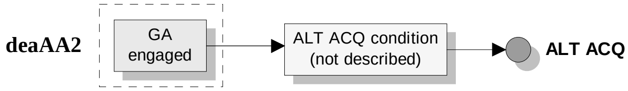

fig. 10. the prototypical representation of the rule $devAA_2$.

Another interesting example concerns the *uncommanded* mode reversions. Mode reversions are uncommanded transitions that occur to preserve the aircraft from hazardous or risky situations. They are similar to the active protections found on some of the Airbus aircraft. Three mode reversions exist on the B737-EFIS. They concern the LVL CHG and V/S modes. $deaLC_1$ (see figure 11) is an example. This uncommanded transition is triggered when the aircraft is engaged in V/S but is unable to achieve one of the targets assigned to the mode: the airspeed is decreasing and drops below a threshold value set to 5 knots below the target airspeed (depending on the FMC update version). To prevent the airspeed from reaching dangerous values (e.g. stall speed), the LVL CHG mode is autonomously substituted to V/S. Since LVL CHG does not have to achieve a specific target vertical speed, the airspeed starts to increase towards the selected airspeed as soon as the mode is engaged.

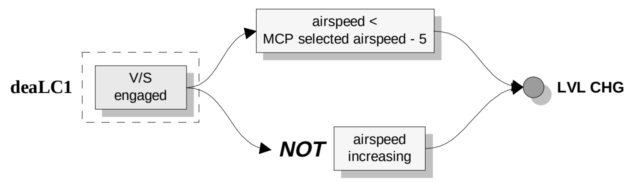

fig. 11. performance limit reversion
(uncommanded transition from V/S to LVL CHG)

The frequency of the rule $deaLC_1$is low. Performance limit reversions are hopefully rare events in normal operational practice, mostly because the two triggering conditions are seldom simultaneously verified (the frequency of their conjunctions is extremely low). As a consequence, the trace of the rule in long-term memory is weak and the rule as a whole is therefore unknown to some pilots. The details of the conditions cannot be recalled exactly when requested to do so. As a result, the transition can occur in operational settings without any detection by the pilot because it was not expected. And when detection occurs, it produces the automation surprises described in the literature (e.g. Palmer, 1995).

6.2. Inferential simplification

A second simplification mechanism is responsible for some of the other knowledge gaps. It derives from the ability of the human cognitive processing system to abstract regular patterns by an *inferential* process (hence the name of the mechanism). As we have seen, the pilot is faced with a rather large set of transition rules that must be stored in long-term memory. The number of rules is huge: forty-four transitions for the four modes studied on the B737-EFIS. Since the aircraft presents many different modes, one can estimate to more than one or even two hundred the number of rules that must be stored in long-term memory. The function of the *inferential* or *abstraction mechanism* is to search for regularities in the rules in order to improve their storage and minimize the memory load. The inferential mechanism has therefore its advantages: the rules that share some features – or conditions – with the rules already stored in long-term memory are easier to learn. But it also has its drawbacks: the rules that do not conform to the general scheme are learned with more difficulties and they are stored in a simplified or distorted form, derived from the patterns already abstracted (over-generalization of the abstract rule).

Several examples of abstract rules exist on the B737-EFIS. The rule AR_1 (fig. 12) is one of them. This *commanded* abstract transition states that any pitch mode can be disengaged if enough force is exerted on the control column. The pitch channel is overridden and the current pitch mode is disengaged and replaced by the CWS P mode. AR_1 is an abstraction of the rules $edvAH_3$, $edvAA_3$, $edvLC_3$ and $edvVS_3$ that correspond to the disengagement by pitch override of the ALT HOLD, ALT ACQ, LVL CHG and V/S modes. The benefit of a generalized abstract rule is obvious: the engagement of CWS P can be triggered by the same action, without any concern for the pitch mode currently engaged. The advantage in term of storage is also clear: a single (abstract) rule is stored in place of the four needed without the inferential mechanism.

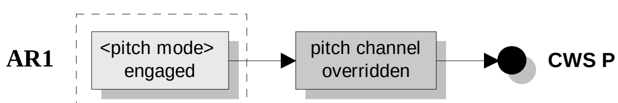

fig. 12. the abstract rule for the pitch override.

Abstract rules also exist for *uncommanded* mode transitions. A very similar example concerns the ability of G/S to disengage the other pitch modes: G/S will engage when the conditions for its engagement are met, whatever the status of the other pitch modes. Because there is a single set of conditions for five different contexts (one for each of the pitch modes), a single abstract rule can be learned. The generation of the action plan (see 6.3 below) is simplified, as well as its execution (the actions to execute do not depend on the active pitch mode). The benefit of an abstract rule here again is obvious.

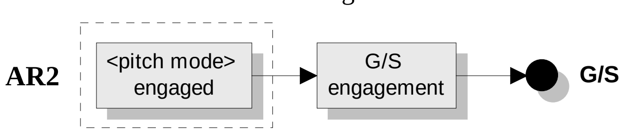

fig. 13. the abstract rule for G/S engagement.

While very useful, the abstraction mechanism can be misleading and contribute to the introduction of knowledge gaps and the production of errors. This is clearly demonstrated by the following example. It involves the termination of (some of) the pitch modes by the *uncommanded* engagement of ALT ACQ. The abstract rule AR_3 presented in the figure 14 states that (some of the) pitch modes can be autonomously disengaged and replaced by ALT ACQ when a specific condition (which is not described in the Operations Manual) is achieved (the transition to ALT ACQ actually occurs when the aircraft approaches from the target altitude set on the MCP panel).

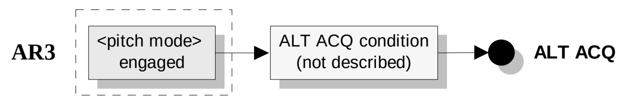

fig. 14. the abstract rule for ALT ACQ engagement

Problems arise here because the rule AR_3 is only partially correct. Contrary to the rules AR_1 and AR_2, AR_3 is true for only a small subset of the pitch modes: LVL CHG, V/S and TO. Substituting any of them to the <pitch mode> variable in AR_3 will produce a rule that actually belongs to the transition rules of the aircraft. Problems – and errors – occurs when the pilots inappropriately apply AR_3 to GA. The explanation for this *generalization* lies in the similarities that exist between GA and the modes to which AR_3 really applies: GA is a pitch mode, it is very similar to TO and it can indeed be terminated by a transition towards ALT ACQ. An aggravating factor is the prototypical form produced by the frequential mechanism on the rule $deaAA_2$ (see fig. 10). It is clearly identical to the rule obtained when instantiating the <pitch mode> variable with GA in the rule AR_3 (see fig. 15) ! The frequential and inferential simplification mechanisms combine their effects to produce the (erroneous) representations of the figures 10 and 15. This is particularly unfortunate, especially if one considers the weakness of the memory trace of $deaAA_2$, (the frequency of this rule is very low). As a general rule, to avoid this kind of pitfall, one should always define abstract rules that apply solely to modes that exhibit some similar external behaviors (e.g. to all the pitch modes, or to all the modes that capture a target value,…). See Javaux (1997) for a discussion on the need to introduce more coherence between the ‘internal’ and ‘external’ behaviors of the modes.

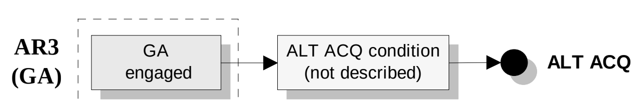

fig. 15. the instantiation of the rule AR_3 for the GA mode

6.3. Plan generation and rule chaining

The third mechanism does not produce a simplification of the representation of the rules. It concerns the *plan generation* mechanism, by which an action plan is produced by the pilot, either to trigger a commanded mode transition or to monitor an uncommanded mode transition. To understand this mechanism, we will consider the following (quite common) situation.

> *The aircraft is flying level (with ALT HOLD engaged) at (the MCP selected altitude of) 6000 ft and is requested by the ATC to descend to 4000 ft.*

The best way to comply with this request (in this specific context) is to use the LVL CHG (level change) mode and trigger the devLC$_1$ commanded transition (fig. 16).

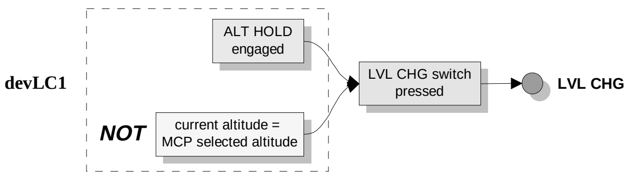

fig. 16. the rule devLC$_1$ for engaging LVL CHG.

To produce the action plan that will actually trigger the transition, the pilot first instantiates the rule in the current operational context. An *instantiation* of devLC$_1$ in the current context (ALT HOLD engaged, current altitude = MCP selected altitude) is obtained by removing from the rule all the preconditions and conditions that are already true (i.e. ALT HOLD engaged). The instance of devLC1 in the current context (identified as Ca) is shown in the figure 17.

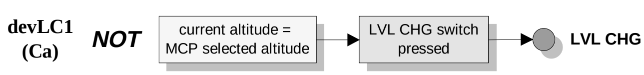

fig. 17. the instantiation of devLC1 in the context Ca.

The instance in the figure only contains the preconditions and conditions that are false in the current context. The following step is to determine which preconditions and conditions are pilot-dependent and which are pilot-independent. Pilot-dependent conditions are the conditions whose truth value can be changed by an action (e.g. press a switch). Pilot-independent conditions change autonomously (e.g. when the aircraft reaches the MCP selected altitude). In the instance presented in the figure 17, the two remaining conditions are pilot-dependent. Their status can be modified by two actions: changing the MCP selected altitude and pressing the LVL CHG switch. This is the action plan the pilot will execute to engage the LVL CHG mode (fig. 18).

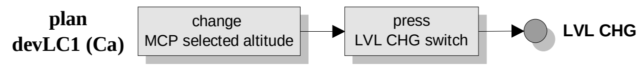

fig.18. the action plan for triggering devLC$_1$ in the context Ca.

The action plan produced by the instantiation mechanism includes the actions the pilot will issue on the man-machine interface to change the pilot-dependent conditions, but also the actions for monitoring the pilot-independent conditions.

Things were simple and straightforward in the example above because the conditions in the plan were *terminal* or elementary conditions. They cannot be further decomposed into simpler conditions. Problems arise when the conditions in the plan are not terminal conditions. This is demonstrated by the following example (fig. 19). It involves a *rule chaining* mechanism that actually increases the cognitive complexity of the plan generation. The example concerns once again G/S and the rule devAH$_1$ already mentioned above (the rule is weakened by the frequential simplification mechanism, see the fig. 7 and 8).

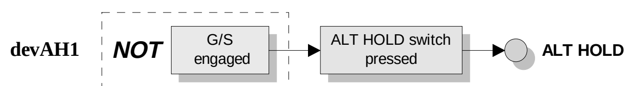

fig. 19. the representation of the rule devAH$_1$.

The difficulties with devAH$_1$ appear in the contexts where the instances of the rule are not solely composed of terminal actions. This is the case when G/S is engaged: in this context (Cb), the action plan presented in the figure 20 is produced. It states that G/S must be first disengaged and then the ALT HOLD switch pressed for triggering the transition.

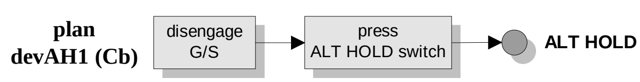

fig. 20. the action plan for triggering devAH$_1$ in the context Cb.

Problems arise here because the action to disengage G/S is not a *terminal action*. It is itself a mode transition! And it involves the satisfaction of a series of other pilot-dependent and –independent conditions before G/S can be truly disengaged. In the context Cb, the action plan for engaging ALT HOLD has therefore two hierarchical levels and involves (at least) two rules. The cognitive complexity of the plan generation is increased by the necessity to coordinate (chain) the execution of the two rules and to maintain some extra information in working memory (i.e. to press ALT HOLD after G/S has been disengaged). This second example related to $devAH_1$ clearly explain why so many pilots have difficulties (and knowledge gaps) with this rule as observed by Sarter & Woods (see the example 2.1 above): the detrimental effects of the frequential simplification and rule chaining mechanisms are combined on a single rule.

Other problems or oddities observed in the simulator can be explained by the rule chaining mechanism (e.g. inefficient action plans produced by some – novice – pilots when engaging the V/S mode). These aspects will not be detailed here.

6.4. The characteristics of the man-machine interface

The performance of the pilots applying the transition rules is not solely determined by the nature of the conditions and their inter-relationships. The rules are situated in the context of the man-machine interface on the flight deck. The objective of this section is analyze how the rules and their conditions interface with the cockpit and in particular how errors can occur at the execution stage, when an action plan has been generated. While the problems described here do not originate from simplification mechanisms and knowledge gaps, they can however contribute to the production of errors with very severe consequences (e.g. the Mont St-Odile and Nagoya accidents, both in 1992).

The man-machine interface is seen as composed of *interactional objects*. The interactional objects contain the objects that *display* information (displays, indicators, lights,…) and the objects that allow the *control* of the aircraft and the automation (controls, commands,…). Their purpose is to allow the interaction with the aircraft and the automation, either when monitoring the status of pilot-independent conditions or when modifying the status of pilot-dependent conditions. Each condition in a transition rule generally involves one or more interactional objects. Terminal pilot-dependent conditions (actions) for example usually involve a single control (e.g. the MCP altitude selector). Terminal pilot-independent conditions typically involve several redundant display elements (e.g. an FMA indication and a light bulb on the MCP panel).

The *display* elements must allow to evaluate the status of the pilot-dependent and pilot-independent conditions, either for producing an instantiation of the rule during the generation of the plan or during the execution stage, when terminal actions are executed and pilot-independent conditions monitored. As a general rule, to reduce the occurrence of errors, there should be a strong relationship between the type and frequency of the conditions and the salience of their associated interactional objects. Pilot-independent conditions should be supported by rather salient interactional objects since their evaluation is critical for the prediction of uncommanded mode transitions (their status changes autonomously). The feedback on the conditions with very low frequencies (they are false most of the time when the rule is applied) should be particularly salient when the condition is true. The feedback on the conditions with very high frequencies (they are true most of the time when the rule is applied) should be particularly salient when the condition is false. In particular, negative feedback (removal of a positive information when the condition is false) should always be rejected.

Some of these recommendations are violated on the B737-EFIS. The feedback on the stab trim condition (a high frequency condition) in the rule $deaAA_2$ (fig. 9) conforms to one of the recommendations suggested above (a positive information in the form of a steady red light, presented to the pilot when the condition is false). The salience of the feedback however is not sufficient because the light is steady and shown in the peripheral vision field of the pilot, in a phase of the flight (during a go-around) where most of the visual attention is focused on a single display, the EADI. This combines to the rather meaningless content of the signal (a red light) which renders the interpretation of a stab out-of-trim condition rather difficult. All this adds to the difficulties already experienced with this rule because of the combination of the frequential and inferential simplification mechanisms mentioned above (see the figures 10 and 15).

The *controls* on the other hand must allow the easy and reliable implementation of the action plans generated after the instantiation stage. The main source of errors at the execution stage are the rather famous (or infamous…) slips of actions. Several types of slips have been described by Reason (1990) and Norman (1981, 1988). The typical slip associated with the interactional objects is the description error (Norman, 1988). It occurs when the correct action is performed on the wrong object. This happens on the flight-deck when the pilot manipulates the wrong switch or knob. Typical – and frequent – slips on the B737-EFIS involve a confusion between the speed and heading rotating knobs on the MCP panel.

7. Discussion and conclusions

A first preliminary remark regarding this research. The study is performed on the B737-EFIS, and it shows - as already demonstrated by Sarter and Woods (1992) - that this aircraft presents some examples of pilot-automation interaction difficulties. This is *not peculiar* to this aircraft or to this manufacturer. Similar results would be found on commercial aircraft made by other manufacturers (e.g. Airbus and McDonnel Douglas). Sarter & Woods (1995) report for example similar pilot-automation problems on the Airbus A320. It is also believed that the errors and knowledge gaps reported in this paper are *not characteristic* of the pilots at Sabena. Similar problems would be observed in the other state-of-the-art companies.

The paper suggests a possible methodology for explaining pilot-automation difficulties on modern, highly-computerized aircraft. The objective is to *predict* where difficulties, errors and knowledge gaps are likely to appear for a given man-machine system, with a specific set of structural and functional properties. The concept of cognitive complexity – derived from the notion of cognitive resources – is proposed. Seven different sub-tasks involved in the interaction with the modes of the automation are dissociated. Two of them related to commanded and uncommanded mode transitions have been studied in great details.

7.1. Commanded and uncommanded mode transitions

The commanded and uncommanded mode transitions for four of the modes of the B737-EFIS have been thoroughly analyzed. Forty-four transitions have been elicited and described in a Boolean and a procedural formalisms. Both formalisms have proved satisfying for the task at hand.

A measure of cognitive complexity has been proposed. The *syntactical* measure of *complexity* attempts to detect the transitions that request or involve the most cognitive resources. It demonstrates that metrical measures of cognitive complexity are feasible and usable (e.g. for the certification process). The interest of this measure is however limited. It does not predict the type of problems, errors and knowledge gaps that are likely to appear with the rules whose cognitive complexity is especially high. Its predictive value regarding performance is therefore limited. A part of the explanation lies in the local character of this measure: the interactions between the rules (e.g. the similarities) are not taken into account.

Four mechanisms or contributing factors have been proposed to address the issue of *performance prediction*: frequential simplification, inferential simplification, rule chaining and the characteristics of the interactional objects. They all attempt to account for – and predict – the errors and knowledge gaps observed in the population of pilots. The four mechanisms or factors contribute to the production of errors at different stages of cognitive processing (fig. 21): the frequential and inferential simplification mechanisms produce knowledge gaps that affect the rules during their storage and maintenance in long-term memory. The rule chaining mechanism leads to the production of errors during the generation of the action plans derived from the rules. And the characteristics of the interactional objects mostly influence the production of errors during the execution stage, when the action plans are performed in the cockpit.

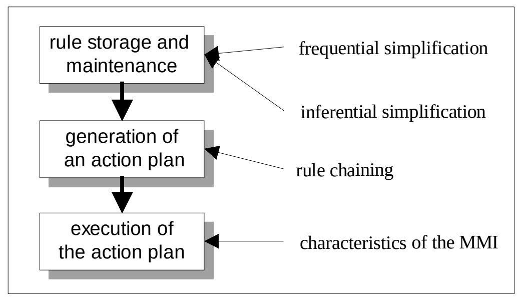

fig. 21. the mechanisms/factors and the information processing stages

The relationship between the four mechanisms or factors and the notion of cognitive complexity is strong. The frequential and inferential simplification mechanisms demonstrate the natural tendency of the human cognitive processing system to reduce the cognitive complexity of the material that must be stored and maintained in memory. The introduction of rules that must be chained obviously increases the cognitive complexity of the interaction, mostly because more rules have to be manipulated and more information maintained in working memory during the generation of the action plan. Finally, a poor interfacing of the interactional objects induces increased attentional loads for monitoring the changing conditions (displays) and for performing the actions without errors (slips).

The set of forty-four transition rules has been confronted to the four mechanisms and factors. The results obtained so far are preliminary and will have to be confirmed by a series of experiments and questionnaires. They are however already considered as very promising because *most if not all* of the difficulties reported by the instructors or acknowledged in the literature have been explained. In several cases, some predictions have been made and the associated difficulties, errors or knowledge gaps, while initially unreported, have been confirmed by the instructors at Sabena.

The continuation of this research involves a series of efforts in several directions:

- to improve the understanding of the cognitive processes involved in the four mechanisms and factors: implicit learning (frequential simplification), categorization and abstraction (inferential simplification), reasoning and short-term planning (rule chaining), slips errors and activation theory (interactional objects).
- to devise an algorithmic method for predicting where errors and knowledge gaps are likely to appear. This consist in an improvement and a systematization of the method applied in this paper. Methods than search extensively the context space for configurations of conditions that increase the cognitive complexity of some rules are considered as a priority.
- to extend the Boolean and procedural descriptions to the other modes on the B737-EFIS.
- to measure the rule and conditional frequencies for the complete set of transitions. A subjective evaluation questionnaire will be submitted to 5 or 6 pilot-instructors at Sabena.
- to test the predictions of the algorithmic method against experimental data. Multiple-choice questionnaire will be used to investigate the knowledge gaps. Simulator experiments will be conducted to verify the predictions about the errors.

7.2. The cognitive complexity approach

The promising results obtained with the commanded and uncommanded mode transitions suggest that it is actually possible to predict some aspects of the pilot’s performance from a detailed characterization of the automation properties. It is believed that the same approach can be applied successfully to the other cognitive sub-tasks involved in the interaction with the automation. Future plans involve the definition of a measure of complexity and the development of predictive methods analog to the ones described above for the sub-task 5 (the interpretation of the Flight Mode Annunciations).

References

Alliot, J.M. & Schiex, T. (1993). *Intelligence Artificielle et Informatique Théorique*. Cépadues Editions : Toulouse.

Billings, C.E. (1997). *Aviation Automation. The Search for a Human-Centered Approach*. Lawrence Erlbaum Associates : Mahwah, NJ.

Degani, A., Shafto, M. & Kirlik, A. (1995). Mode Usage in Automated Cockpits: Some Initial Observations. In T.B. Sheridan (Ed.), *Proceedings of the International Federation of Automatic Control; Man-Machine Systems (IFAC-MMS) Conference*. IFAC: Boston, MA.

Degani, A. (1996). *Modeling Human-Machine Systems: On Modes, Error, and Patterns of Interaction*. Ph. D. thesis, Georgia Institute of Technology.

Degani, A., Shafto, M. & Kirlik, A. (1998). Modes in Human-Machine Systems: Review, Classification, and Application. Submitted to *The International Journal of Aviation Psychology*.

Javaux, D. (1997). Measuring Cognitive Complexity in Glass-Cockpits. A Generic Framework and its Application to Autopilots and their Modes. In R. Jensen & L. Rakovan (eds), *Proceedings of the Ninth Symposium on Aviation Psychology*, Columbus, Ohio.

Javaux, D. & De Keyser, V. (1997). Complexity and its Certification in Aeronautics. In *Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, October 12-15, Orlando, FL.

Kieras, D.E. & Polson, P.G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22:365-394.

Leveson, N.G. & Palmer, E.A. (1997). Designing Automation to Reduce Operator Errors. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL.

Monnier, A. (1993). *Rapport de la Commission d'enquête administrative sur l'accident du Mont Saint-Odile du 20 janvier 1992*. Ministère de l'Equipement, des Transports et du Tourisme : Paris, France.

Norman, D.A. (1981). Categorization of Action Slips. *Psychological Review*, 1 (88), 1-15.

Norman, D.A. (1988). *The Psychology of Everyday Things*. Basic Books.

Palmer, E. (1995). Oops, "It Didn’t Arm." A Case Study of Two Automation Surprises. In R. S. Jensen & L.A.

Rakovan (Eds.), *Proceedings of the Eighth International Symposium on Aviation Psychology*, pp. 227-232, April 24-27, Columbus: Ohio.

Reason, J. (1990). *Human Error*. Cambridge University Press: New York.

Sarter, N. & Woods, D.D. (1992). Pilot interaction with cockpit automation. I. Operational experiences with the Flight Management System. *International Journal of Aviation Psychology*, 2, 303-321.

Sarter, N. & Woods, D.D. (1994). Pilot interaction with cockpit automation. II. An experimental study of pilot's mental model and awareness of the Flight Management System (FMS). *International Journal of Aviation Psychology*, 4 (1), 1-28.

Sarter, N.B. & Woods, D.D. (1995). "Strong, Silent, and 'Out-Of-The-Loop'". *CSEL Report 95-TR-01*. February 95.

Wickens, C.D. (1992). *Engineering Psychology and Human Factors*. Harper Collins : New York.

Wiener, E.L. (1989). *Human factors of advanced technology ("glass-cockpit") transport aircraft*. NASA contractor Rep. N°. 177528. Moffett Field, CA: NASA-Ames Research Center.