To link to this article : DOI :10.1145/2702123.2702521
URL : http://dx.doi.org/10.1145/2702123.2702521

To cite this version: Dehais, Frédéric and Peysakhovich, Vsevolod and Scannella, Sébastien and Fongue, Jennifer and Gateau, Thibault "Automation Surprise" in Aviation. (2015)
In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems CHI 2015, 18 April 2015 - 23 April 2015 (Korea, Republic Of).

"Automation Surprise" in Aviation: Real-Time Solutions

Frederic Dehais[1], **Vsevolod Peysakhovich**[1], **Sebastien Scannella**[1], **Jennifer Fongue**[1], **Thibault Gateau**[1]

[1]Neuroergonomics Lab - ISAE
10 avenue E. Belin
31055 Toulouse, France
{frederic.dehais,vsevolod.peysakhovich,sebastien.scannella,jennifer.fongue,thibault.gateau}@isae.fr

ABSTRACT

Conflicts between the pilot and the automation, when pilots detect but do not understand them, cause "automation surprise" situations and jeopardize flight safety. We conducted an experiment in a 3-axis motion flight simulator with 16 pilots equipped with an eye-tracker to analyze their behavior and eye movements during the occurrence of such a situation. The results revealed that this conflict engages participant's attentional abilities resulting in excessive and inefficient visual search patterns. This experiment confirmed the crucial need to design solutions for detecting the occurrence of conflictual situations and to assist the pilots. We therefore proposed an approach to formally identify the occurrence of "automation surprise" conflicts based on the analysis of "silent mode changes" of the autopilot. A demonstrator was implemented and allowed for the automatic trigger of messages in the cockpit that explains the autopilot behavior. We implemented a real-time demonstrator that was tested as a proof-of-concept with 7 subjects facing 3 different conflicts with automation. The results shown the efficacy of this approach which could be implemented in existing cockpits.

Author Keywords

Automation surprise, Attentional process, Eye tracking, Human-computer interaction, Formal conflict detection

ACM Classification Keywords

H.1.2

INTRODUCTION

The integration of automation began in the early 1960s with systems that aimed at stabilizing the aircraft through a mechanical manipulation of the flight control surfaces. Automation technology continues to develop and become more and more integrated in aviation thanks to the creation of digital computers and the desire to render aircraft more reliable, more precise, safer, more economical and more productive. Following such developments, the accident rate dropped from more than 35 accidents per one million flight to finally reach the actual one of 0.66 accidents per one million flights [2], an unprecedented improvement in aviation safety records.

This improvement in safety records was unquestionably driven by the development of automation in aviation. For example, Breen [3] showed that the introduction of ground proximity warning systems led to a decrease in the controlled flight into terrain accident rate. Norman [25] supported and generalized this idea by stating that human performance was aided by the improvements in aviation safety resulting from the integration of automation. However, the accident rate never managed to reach the perfect zero accident per million flights, as it has been shown that the tendency is stagnation at the rate reached in the 1980s [2].

While automation helped make aviation safer, its integration in civil commercial aviation flights was immediately followed by a series of crashes due to unexpected problems. For example, in 1972, a L-1011 crashed into the Florida Everglades after a disconnection of an autopilot function while the pilots were dealing with a landing gear problem [1]. This event, among others, caused great worry among human factors researchers in the 1980s. In 1989, the Air Transport Association (ATA), showed its concern towards these issues: *"Although many of these benefits have been realized, serious questions have arisen and incidents/accidents have occurred which question the underlying assumptions that the maximum available automation is ALWAYS appropriate or that we understand how to design automated systems so that they are fully compatible with the capabilities and limitations of the humans in the system"*.

Following this report, human factors issues and the interaction between automation and human have been studied. It was identified that the link between the operator and the automated system was the fragile segment of the overall system [34]. Indeed, it was shown that the human operator had not been taken as an element of the overall system when designing automated systems, thus, these problems were the result of *"a failure to design for a coordinated team effort across human and machine agents as one cooperative system"* [34]. However, despite the research aimed to design automated systems and training their operators, automation-related problems persisted.

More recently, on June 2009, Air France flight AF447 crashed in international waters of the Atlantic Ocean while flying from Rio de Janeiro to Paris. The French safety board for accident investigation found that the inadvertent icing of the pitot tubes led to abnormal speed measures and false indications in the cockpit [4]. Erroneously believing the aircraft in stall, the autopilot was automatically self-disconnected. The pilots tried to carried out the standard procedure following the disconnection of the autopilot. However, the surprise resulting from the autopilot's disconnection, the inexperience of the pilots, and multiple other causal and contributory factors led the aircraft to actually stall without enabling the crew to manage it. This recent failure by the crew of a well-renowned company in managing an automation-related incident shows that even after 40 years of utilisation of automation in civil commercial aviation, human factors issues linked to automation are still a significant safety concern.

The lesson to be learnt from these unexpected problems and human factors research is that the relationship between the pilot and the autopilot of civil aviation aircraft is prone to conflicts. Studies towards the development of a predictive pilot-automation conflict tool should have to be conducted, and could result in an appropriate solution to help solving and predicting conflicts. In order to accomplish this general objective, we propose to conduct in the next section a further review of the literature to better understand pilot-autopilot cooperation breakdown.

Conflict with automation: "automation surprises"

The analysis of air safety accidents and experiments conducted in flight simulators [14] revealed that the occurrence of a conflict impairs human operators performance. Conflicts are defined as the impossibility for one agent (e.g. a pilot) to reach a goal that matters [6, 12, 16]. This impossibility may be the consequence of limited resources, incoherent data in the user interface, or a contradictory action of another agent such as a crew member or the autopilot [12]. Indeed, conflicts do not solely occur between humans operators, but also involve pilots and automation [12, 39]. Such conflicts are likely to occur as pilots and automation, which have different "logic", may take over from each other on flight guidance. Therefore, it is worth noting that the understanding and the use of modern autopilots requires the crew to master its complex behavior (e.g. up to 42 states and 268 states transitions for the Airbus 340). Some authors have postulated that insufficient pilot mental models of the automation and poor automation feedback [8, 20] can lead to the following conflicting situations:

- "Indirect mode changes" [21] or "automated behavior" [17]: automation automatically changes its behavior in absence of any pilot's action (e.g. automatic go-around during landing, independently of the pilot's will);

- "Unintended side effects" [21]: a single pilot's action triggers a cascade of autopilot state changes (e. g. an action on the horizontal guidance mode modifies automatically the autopilot vertical guidance behavior);

- "Inhibited behavior" [17] or "operator authority limits" [21]: the pilot's actions are automatically "cancelled" by the system and have no effect on autopilot behavior.

Conflicts are not limited to their structural/formal aspects (i.e. the impossibility for the pilot to reach his initial goal) but also share psychological and cognitive attributes. Conflicts with automation can remain unnoticed and can lead to erroneous situation awareness ("mode error") [15, 19, 33] or attentional tunneling [12] as revealed by eye tracking measurements. Conflicts with automation, when detected but not understood, may also provoke *"breakdowns in the interaction between human operators and automated systems. The result can be situations where the operator is surprised by the behavior of the automation"* [34]. Sarter, Woods, and Billings [34] defined this conflicting situation as "automation surprise". For instance, the concept of opacity of automation was introduced to illustrate that automation surprise leads pilots to questioning the system's behavior [38]: What is it doing now? Why did it do that? What is it going to do next? It was postulated that occurrence of such situation could summon up pilots cognitive resources leading the crew to persist in solving a minor conflict with automation [1] *"instead of switching to another means or a more direct means to accomplish their flight path management goals"* [40]. Though users' surprise and confusion phenomenons were largely addressed using eye tracking for usability studies (see [28] for a good review), no such research was achieved in the aviation domain. Indeed, to the authors' knowledge, "automation surprise" has mainly led to conceptual studies, and no studies in flight simulators with objective measurements have been conducted. We propose to fill this gap by performing an eye-tracking study to better understand pilot's cognitive performance when facing such a situation and pave the way to design assistance tools.

Present study

The general objectives of this research are:

- To contribute to research in solving automation surprises problems in aviation;

- To evaluate the impact of automation surprise on pilot's eye movements as an indirect measure of cognitive performance [28];

- To develop and to test real-time solutions that could be implemented in existing cockpit.

To meet these goals we first designed an experimental scenario involving a conflict with automation. This event was designed such that it would be easily detected by pilots and thus provoked "automation surprise". The objective of this first experiment was to collect behavioral data, especially eye tracking data, to analyze pilot's cognitive performance when facing such a situation. We then proposed an approach to formally detect conflicts and to automatically trigger auditory messages to help pilots solve them. These solutions were implemented in real-time in our flight simulator and led to a proof of concept testing.

AUTOMATION SURPRISE EXPERIMENT

Material and Methods

Participants

Sixteen healthy volunteers (two females; mean age = 30.8, SD = 14; mean flight experience = 1391 hours, range 55–9000; mean automated flight desk experience 200 hours, range 15–700), all pilots from the French Aerospace engineering School (ISAE) participated in this experiment after giving written informed consent. Participants had good auditory acuity and normal or corrected-to-normal vision according to their pilot medical certificate.

Flight simulator

The experiment was conducted with the ISAE flight simulator (figure 1). The user interface was composed of the classical electronic flight instrument displays including the primary flight display (PFD) and the navigation display. The participants had a side-stick, rudder pedals, thrust levers and a Flight Control Unit (FCU) to control the flight guidance. The FCU was dedicated to interacting with the autopilot via four knobs (speed, heading, altitude, vertical speed). The autopilot (vertical and horizontal profile management) was engaged and disconnected via a push button "AP" on the FCU. Autopilot disconnection was associated with a dedicated auditory warning ("cavalry charge"). The auto-thrust (speed management) was engaged and disengaged via a push button "ATHR" on the FCU.

Figure 1: ISAE 3-axis motion flight simulator. FCU: Flight Control Unit. PFD: Primary Flight Display. ND: Navigation Display.

For the purpose of the experiment, we designed a realistic auto-flight system that replicated poor automation designs from different civilian transportation aircraft (see [27] for a formal description of our autopilot).

- The autopilot had one lateral mode ("Heading") and three vertical modes ("positive vertical speed", "negative vertical speed", "null vertical speed"). Note that the vertical speed was null (i.e. 0 ft/min) when the autopilot reached the target altitude or when the pilot pushed the vertical speed knob to level off. These different flight modes were displayed on the upper part of the PFD;
- The autopilot automatically disconnected during overspeed or low speed/stall events. The overspeed value depends on three possible flaps configurations 180 knots, 220 knots and 330 knots respectively. These events triggered the speed auditory warning (triple chime) that inhibited the autopilot disconnection auditory warning;
- The aircraft leveled off in case of inconsistent programming of the vertical speed with regards to the target altitude (i.e. it was not possible to "climb" with a negative vertical speed or to descend with a positive vertical speed);
- The autopilot reversed from level off or negative vertical speed to positive vertical speed when the aircraft speed was 5 knots below overspeed. This mode reversion was dedicated to avoid a possible overspeed.

Experimental scenario

The scenario, which included the occurrence of one conflict with automation, lasted 10 minutes. The simulated ATC (Air Traffic Control) cleared the pilots for take-off from Blagnac airport (Toulouse, France) and the airplane was vectored regularly according to a flight plan that was identical for each participant. At 9000 ft, the ATC required the participant to "accelerate 325 knots, descend 5000 feet with a -1000 ft/min vertical speed" to avoid an incoming aircraft flying at the same flight level. As the speed reached 325 knots, the autopilot reversed to positive vertical speed mode (+1000 ft/min) to anticipate potential overspeed (330 knots - see previous section for autopilot logic). This situation eventually led the airplane to level off instead of descending as the selected target altitude (i.e. 5000 feet) could not be reached with a positive vertical speed (figure 2).

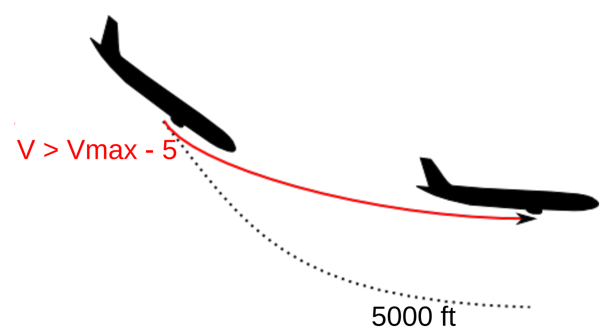

Figure 2: The "impossible descent" automation surprise: the ATC required the participants to descend but with an excessive speed. The combination of two autopilot behaviors led the aircraft to level off to prevent overspeed.

Procedure

The participants had a 20-minute tutorial presentation that detailed the flight simulator user interfaces (PFD, FCU, etc.) with a special focus on the different nominal and off-nominal autopilot behaviors. In order to check the understanding of the tutorial, the participants had to comment on each slide of the tutorial and to recall the autopilot behavior. The volunteers completed a one-hour training session in the flight simulator that included basic manual flying, landings, take-offs, stall and overspeed recovery. They were then trained to interacting with the autopilot and had to set different parameters according to ATC instructions (e.g. "Supaero32, steer 200 degrees, climb 6000 feet, vertical speed +1000 ft/min"). Eventually, off-nominal situations such as autopilot automatic disconnection due to overspeed/stall events, inconsistent FCU programming, level off to vertical speed mode reversion were induced and the participants were told to recover from them. After the training, the participants had to repeat once the interaction with the FCU and the autopilot behavior as well as the different auditory warnings. The experimental scenario was employed after this training.

Behavioral Data Analysis

Flight parameters, lateral and vertical modes of the autopilot, and the pilot's use of the FCU knobs were collected. The volunteers were also debriefed right after the end of the experiment and were told to explain their understanding of the autopilot behavior during the descent and to comment on their own actions. The analysis of these data allowed us to characterize conflict occurrence and conflict solving. As stated in the introduction section, conflict is defined as the impossibility to reach a goal that matters. In aeronautics, goals matter for safety reasons such as avoiding loss of control and collisions with terrain or with an incoming aircraft. In our experimental scenario, we characterized conflict as unsolved when volunteers failed to manage the vertical separation with the incoming traffic according to aeronautical rule (the minimum vertical separation between two aircraft is 500 ft). We characterized conflict as solved when participants performed an action that enabled them to recover from the critical situation (i.e. reducing selected speed with the dedicated FCU knob or manually taking over to continue on descending).

Ocular Data Analysis

Participant eye movements were recorded with a head-mounted Pertech® eye tracker. This lightweight non-intrusive device has 0.25° of accuracy and a 50 Hz sampling rate. After a 9-point calibration, a video capturing the visual scene (eye tracker frontal camera) was available with the eye gaze mapped on it. The vertical and horizontal movements in degrees were also computed.

We detected ocular events of three possible types (blink, saccade and fixation) with a dispersion-velocity based algorithm. Gaze moving under 30°/s with a dispersion threshold of 1° were considered as fixations. Other samples (if the eye was not closed) were considered as saccades. The number of occurrences of these ocular events can be used to characterize the visual processing behaviors. Thus, saccades of small amplitudes characterize local information search whereas those of large amplitudes characterize global information search. As for fixations, we distinguish two types of these ocular events: short ones, the insufficient length of which does not allow the participant to extract and process the associated information and are therefore associated with information search; and long ones, linked to sustained visual and cognitive processing of fixated information. For more detailed discussion on eye-fixation durations see Salthouse et al. [32]. As the minimum time one needs to process basic interface information depends on the interface/environment, the exact separating threshold is questionable. Short fixations are rarely considered as information processing in most of studies where authors prefer to choose a threshold of 200 ms to detect a meaningful fixation [24].

To express the balance between search and extraction/treatment of information, Goldberg and Kotval [18] proposed to aggregate the fixation and saccade rates in a sole ratio. Regis et al. [29] used a modified ratio to identify attentional tunneling (i.e. excessive focus). Therefore, we decided to follow this approach and to search for a particular ocular behavior that would characterize the moment of conflict in comparison to a nominal autopilot monitoring sequence. The autopilot mode change (i.e. "level off" to "positive climb") triggered the onset of conflict. The baseline window was chosen as the onset of conflict window, minus two minutes of flight that brings it to a nominal autopilot monitoring activity (i.e. basic autopilot supervision and interaction with FCU knobs). We chose a 15-second window of basic piloting activity and compared it to the equivalent 15-second window following the rise of the conflict. We ran non-parametric Wilcoxon tests on fixations and saccades of different lengths (multiples of 20 ms – the intersampling period) to determine what events best characterize the conflict occurrence. We selected then all significant events and merged adjacent length-values in the same class. We obtained thus three classes: saccades of 120–160 ms duration, short fixations of 80–120 ms duration and long fixations of 240-260 ms. According to the previously mentioned above literature, we argue that the short fixations are not long enough to extract the complex information displayed in cockpit, and were therefore associated with information search. Keeping these three classes, one can modify the Goldberg and Kotval's idea and compose a new adapted "explore/exploit" ratio defined as

$$R = \frac{\text{saccades} + \text{short fixations}}{\text{long fixations}}.$$

On one hand, when participants explore the interface, the number of saccades and short fixations increases while the number of long fixations decreases; leading to an increase of the ratio. On the other hand, when participants exploit the interface, long fixations dominate, and the ratio decreases. Figure 5 illustrates the evolution of the ratio on three most representative participants.

Results

The debriefing session revealed that 14 out of 16 pilots perceived the abnormal behavior of the autopilot. Among them, we found that only 7 out of 16 pilots understood the situation and initiated the correct recovery procedure within the critical time window of 30 s to avoid the incoming traffic (mean reaction time 18 s after the onset of the automation surprise, SD= 9.3). Regarding the Wilcoxon Matched pairs tests over the eye events, we found significant effects of the conflict corresponding to an increase of the saccade rate ($p < 0.05; Z^1 = 2.04$) and short fixations ($p < 0.05; Z = 2.36$) and a decrease in the long fixation rate ($p < 0.001; Z = 3.40$); (Fig. 3). In addition, the "explore/exploit" ratio over the 16 pilots (Fig. 4) revealed a significant effect of the conflict with higher ratio for the conflict situation comparing to the "baseline" ($p < 0.001; Z = 3.29$) confirming that the conflict implies higher exploration activity. In addition, a complementary analysis over the ratio did not reveal statistical difference between the group of pilots who solved the conflict and the group of pilots who did not.

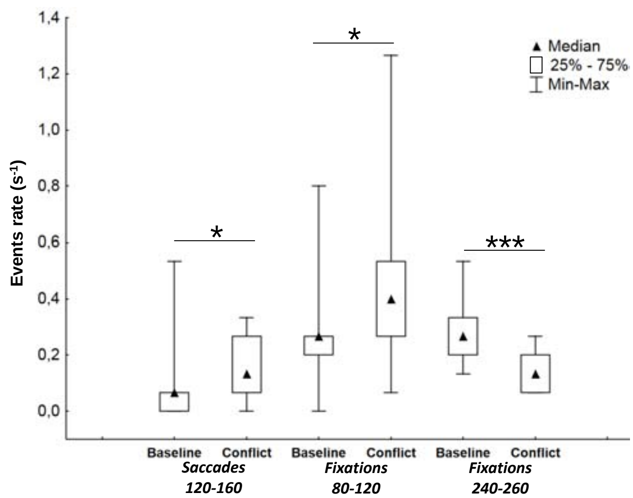

Figure 3: Group results (N=16) over the eye events rate. Ocular data group results. *:$p < 0.05$; ***:$p < 0.001$.

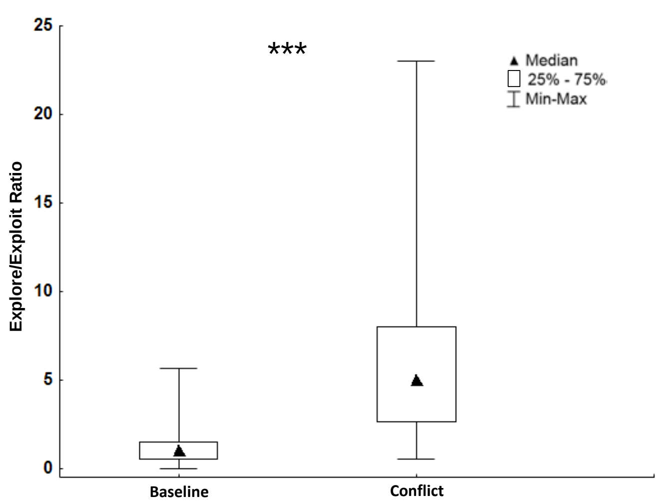

Figure 4: Group results (N=16) of the"Explore/exploit" ratio. The conflict situation leads to an increase in the ratio value. Ocular data group results. *:$p < 0.05$; ***:$p < 0.001$.

[1]Please note that Z is a converted "W" of the Wilcoxon test.

Conclusion

The objective of this experiment was to induce a conflict with automation that would provoke "automation surprise". Fourteen pilots out of 16 declared that they perceived a conflicting situation as the automation prevented them from descending to avoid a collision with an incoming aircraft. Whereas conflict solving was "straightforward (i.e. reducing the selected speed with the dedicated FCU knob), most of pilots were stuck and failed to deal with the situation immediately. Many participants made typical "fixation errors" [9] as they persisted in disengaging and reengaging the autopilot (i.e. lateral/vertical guidance) or dialing in vain the altitude or vertical speed knobs on the FCU. Only 7 pilots managed to solve the conflict but with a long mean reaction time ($18s$) considering that a collision with another aircraft was possible. Moreover, the analysis of ocular events revealed that the volunteers exhibited higher visual search (more short fixations and saccades) to the detriment of information processing (less fixations) during conflict in comparison to baseline. It is worth noting that the ratio explore/exploit (i.e search $vs.$ process) dramatically increased and was five times higher during conflict than baseline. As the focus of attention is mainly top-down and goal-driven [23], it was not surprising that participants' poor mental model led to erratic eye movements during the conflict. A complementary explanation to this ocular behavior is that these unexpected "automation surprise" situations may have suddenly intensified pilots' mental workload and psychological stress. The occurrence of such stressors are known to negatively impact the triangular circuit of attention (i.e. engage/shift/disengage) promoting either excessive focus [11] or, on the contrary, an inability to focus and to filter out irrelevant information [37]. Indeed, "automation surprise" led to an excessive but inefficient visual search that prevents pilots from extracting the relevant information (i.e. the speed indicator). Taken together, these results tend to show that occurrence of automation surprise event summon up attentional resources toward conflict solving. This is akin with previous studies showing the link between conflicts, attentional impairment [12] and persistence in erroneous course of actions [1, 11]. One coud consider that this experiment was not conducted with airline pilots but with light aircraft pilots. However, we believe in the validity of our results as our population of pilots were trained to deal with critical events (e.g. overspeed) and had significant autoflight system expertise as current light aircraft integrate complex autopilot and electronic displays similar to modern transportation ones.

DETECTING AND SOLVING CONFLICTS

As shown the previous experiment, another step has to be considered to mitigate the occurrence of human automation conflicts in current cockpits. We therefore propose an approach, based on the analysis of flight parameters, pilots' actions and autopilot states, to detect conflicts with automation and to synthesize messages that provide explanations of the autopilot behavior.

Conceptual approach

Many formal studies have proposed to detect conflict with automation using finite state automata (e.g. [8, 13, 20]). Au-

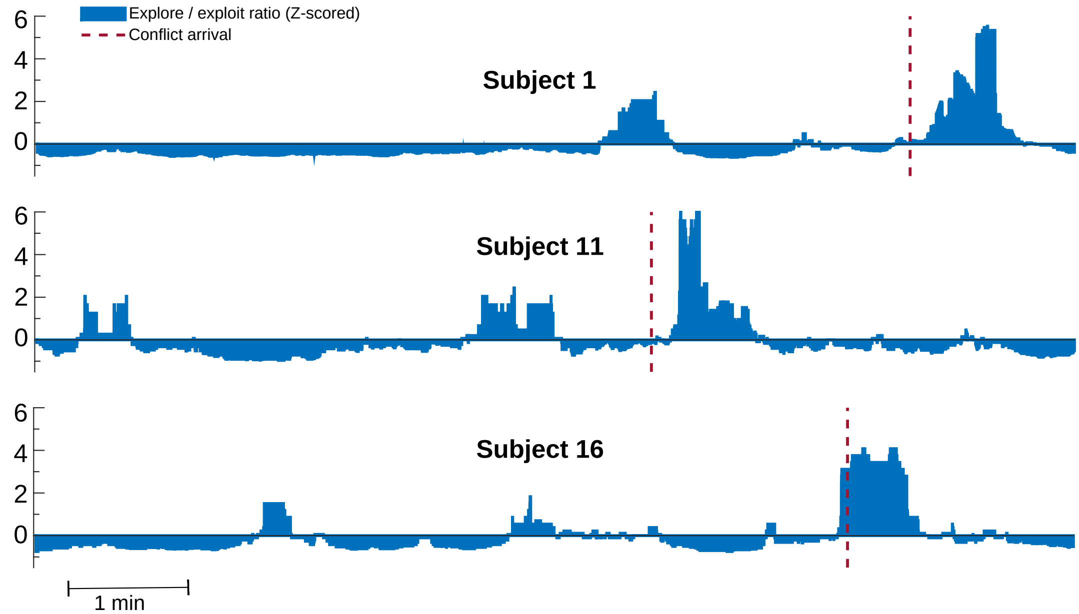

Figure 5: An example of z-scored explore/exploit ratio for three representative participants. The red line shows the formal arrival of conflict.

thors generally implemented model checking techniques to assess the design of pilot-autopilot interaction [5, 22] and to predict automation surprise [8, 20, 30, 31]. However these approaches were implemented off-line and in an ad hoc manner as they required expert knowledge to implement pilots' poor mental model of automation. Our approach differs from previous work: the model is generic and does not represent a specific conflicting situation. Moreover, we do not make any assumptions about mental models to represent pilots imperfect knowledge. This eliminativist position is based on the fact that inter-individual variability of these mental models is a barrier to formalization. Instead, we propose to detect conflict using observable and objective facts. A transition from one automated flight guidance mode to another could result either from a pilot's action (e.g. dialing a FCU knob) or from an autopilot/system event regulating its behavior. Note that the conflicting situations described in the literature (i.e. "indirect mode changes", "unintended side effects", "inhibited behavior" – see introduction) involve the firing of an automated transition (e.g. overspeed) that interferes with the crew's goals [27]. Such automatic "silent transitions" are likely to cause conflicts as the one observed during our first experiment. Therefore, a reasonable approach to detect conflict in real-time is to track autopilot "automated/silent" mode changes that were not provoked by any pilots' actions. Once conflicts are identified, a second challenge is to design solutions to help pilots solving the conflict.

Chialastri [7] stated that *"until the mid-1990's, it appears that pilots tried to adjust their behavior to a given challenge (new automation), which was conceived regardless of their actual need and deeply rooted habits."* Designers have developed various ways of mitigating these situations such as improving flight deck design to provide an efficient communication between the human operator and the aircraft. However, concurrently, problems linked to human limitations of attentional resources also arose: "*With the introduction of the glass cockpit, the traditional flight instrument display was replaced by a different presentation encompassing more information, greater flexibility, color coding and marking, but at the same time, it could lead to information overload, as several parameters are displayed in a compact area*" [7]. As revealed in our experiment pilots experienced excessive visual search and failed to identify relevant information to understand autopilot behavior. Autopilot user interface conveys very poor information that does enable scrutability [36]: it only displays the automation current state (e.g. "climb", "turn") but it does not provide any explanation about its behavior (i.e. what is the reason for state changes). Therefore displaying information about "why this autopilot behavior" could help pilots to solve conflict with automation. This response could be done automatically through the analysis of the history of automation state transitions and events that led to the conflicting situation.

Pilot's assistant system implementation

We propose to design a pilot's assistant system based on the previous considerations. It is composed of two main elements: formal conflict detection and message synthesizer. The following section describes the implementation of these two elements.

The first step is to detect the formal conflict. Inconsistency between the pilot's settings and the autopilot behavior is a key for conflict detection. We suppose that pilot is rational and expresses his intentions trough the FCU interface. We propose thus to compare in real-time pilots' last settings in the FCU and automation state. Differences between these two pieces of information can emphasize a potential conflict. Let V_{PG} be the current goal state of the pilot (deduced from the last FCU inputs); and let V_{AP} be the current state vector of the autopilot. When the autopilot is activated, this last vector represents the instructions sent by the automata to control flight guidance.

We define the *consistency checker* (Fig. 6) – a module that checks in real-time the consistency between the two goal state vectors

$(V_{PG}, V_{AP}) \in (SPD, HDG, ALT, V/S, AP)^2$, where:

- SPD: the speed that can be selected;
- HDG: the heading that can be selected;
- ALT: the altitude that can be selected;
- V/S: the vertical speed that can be selected;
- AP: autopilot button position (On/Off);

The following example illustrates "the impossible descent" conflict (see Fig 2). Before the occurrence of the conflict, we have:

$$V_{PG} = V_{AP} = [325, 140, 9000, -1500, \text{On}].$$

The aircraft reaches a near-overspeed state, which involves autopilot reversion to +1000 ft/min. The autopilot goal vector is thus changed:

$$V_{PG} = [325, 140, 9000, -1500, \text{On}]$$
$$V_{AP} = [325, 140, 9000, \mathbf{+1000}, \text{On}]$$

However, there is an inconsistency as the selected target altitude, which is below the current altitude, cannot be reached with a positive vertical speed. As a matter of fact, the plane leveled-off, instead of descending as initially requested:

$$V_{PG} = [325, 140, 9000, -1500, \text{On}];$$
$$V_{AP} = [325, 140, 9000, \ \mathbf{+0}, \text{On}].$$

We notice that $V_{PG} \neq V_{AP}$, therefore we may have a potentially conflictual situation.

The second step is to automatically generate messages adapted to the current situation. As long as autopilot documentation is available, it is possible to design a finite state machine (FSM) that represents its behavior. Using this formal description, an *autopilot state manager* module (see Fig. 7) is able to track transitions between different states from the autopilot automata. The knowledge of the last transition (initial state, resulting state, type of transition) informs us about what event entailed the conflict.

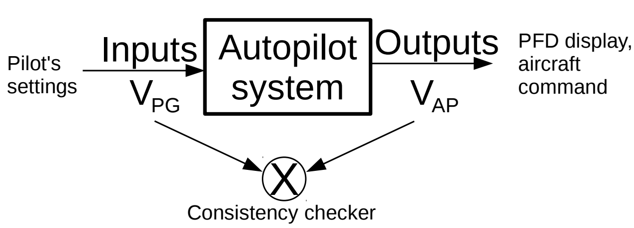

Figure 6: Consistency checker: comparison between pilot's inputs and autopilot outputs.

Thus, we can generate an appropriate message. We defined the following message syntax: "<*current state*> due to <*last transition*>". Pilot is then informed of the current state of the autopilot, and of the transition that leads to the conflictual state. Knowing that, he has many clues to deduce what action must be undertaken. For example, if the autopilot reverses to climb to avoid an overspeed, the message will be "<climbing +1000> due to <near overspeed>". Logically, the pilot will be inclined to reduce speed. We used auditory message for the sake of implementation ease.

Using this two steps, we implemented a way to warn the pilot of an eventual conflictual situation with the autopilot, providing him also with information about the conflict's origin to make easier its resolution (Fig. 7):

- the *consistency checker* identifies potential conflict between pilot and autopilot using pilot's entries and the autopilot flight settings;

- meanwhile the *autopilot state manager* module keeps track of FSM states and transitions. When a potential conflict arises, a message is generated to warn the pilot and to provide him a possible solution.

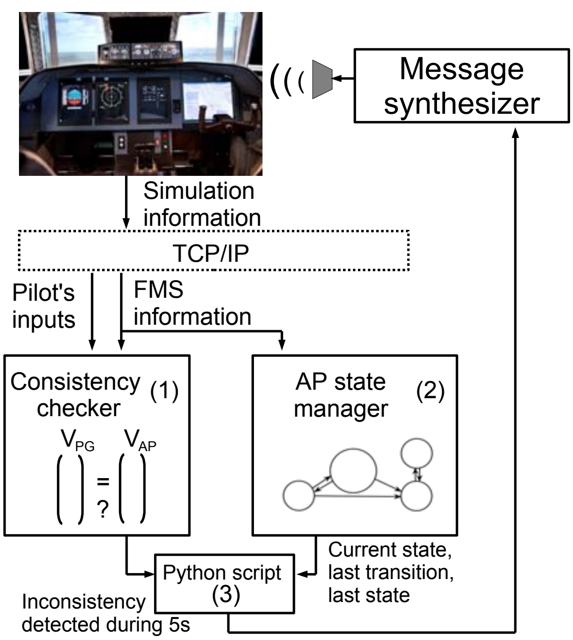

Figure 7: The *Consistency checker* (1) checks for potential conflicts (when autopilot is engaged), while *autopilot state manager* (2) records states and transitions from the autopilot automata. When a potential conflict is detected, and the pilot's actions are not solving it within 5 seconds, a message is triggered (3), using history from *autopilot state manager* to synthesize the correct message for the pilot.

Technically, we used a TCP/IP connection to collect pilot's settings on the autopilot *via* his actions on the four knobs of the FCU (V_{PG}). We inferred autopilot states from the orders sent by the autopilot automata to the aircraft(V_{AP}).

Leaving some time for the pilot to react, we wait 5 seconds before sending the message according to pilot's reaction time in critical situations (e.g. pull up alarm). The algorithm 1 and the FSM model were implemented in Python programming language to illustrate our proof of concept.

Algorithm 1: ASSISTANCEMANAGER

while *AssistanceManager running* **do**
  if *not* V_{AP} ***equals*** V_{PG} **then**
    Wait
    if *not* V_{AP} ***equals*** V_{PG} **then**
      generateMessage(current state,last transition)

Proof-of-concept

Motivation

The purpose of this part of the research was to conduct a technical demonstration to test the efficiency of our algorithm to detect automation surprise situations and to automatically generate assisting solutions. The main objective was to convince automation designers that straightforward solutions could contribute to enhance flight safety. Hence, we did not use eye tracking on purpose as to support evidence that conflict detection does not require the integration of such technology and that our solutions could be easily implemented in current cockpits.

Participants and protocol

Seven new healthy volunteers (all men; mean age 31.0, SD = 12.7; mean flight experience 609.7 hours, range 80–4000; mean automated flight desk experience 150 hours, range 30–200), all French defense staff from ISAE participated in this proof of concept testing after giving written informed consent. Participants had good auditory acuity and normal or corrected-to-normal vision according to their pilots' medical certificate. The tutorial and training were identical to the first experiment procedure. The scenario involved three conflicting situations. The participants were cleared for takeoff from Blagnac airport and at 1000 ft the ATC required them to steer 330 and to accelerate 176 knots.

Conflict 1: At 1200 ft, the ATC requests the participants to level off to avoid an incoming traffic flying at 1800 ft. As the speed reached 5 knots below maximum takeoff speed, the autopilot climbs at +1000 ft/min to avoid possible overspeed. This mode reversion could lead to a collision with the traffic as the aircraft is not leveling off anymore. If the consistency checker detected this, thus the following auditory message would be synthesized: "<climbing at +1000 ft/min> due to <near overspeed>".

Conflict 2: When the aircraft cruises at 1 500 ft, it is cleared to steer 143 degrees and to climb 11 000 ft. Cruising at 10 700 ft, the volunteers faces a strong jet stream provoking a brief overspeed and thus disconnects automatically the autopilot: this situation leads to an overshoot of the target altitude (11 000 ft) as the aircraft keeps on climbing. If the consistency checker detected this, thus the following auditory message would be synthesized: "<autopilot disengaged> due to <overspeed>".

Conflict 3: The third conflict is identical to the "impossible descent" situation used in the "automation surprise" experiment (see Fig. 2). If the consistency checker detected this, thus the following auditory message would be synthesized: "<Level-off> due to <near overspeed>".

Efficiency of the messages was assessed with flight parameters and pilots' actions on the FCU to determine whether the conflicts were solved or not; and a questionnaire determining whether it effectively helped the pilots.

Results and conclusion

The consistency checker detected in real-time the occurrence of 20 conflicts out of 21 (i.e. 3 conflicts $\times$ 7 participants). We did not find the cause of this conflict misidentification possibly due to a temporary connection issue with the flight simulator. All of the pilots managed to solve each conflict in time with respect with the 500 ft separation rule. The debriefing analysis confirmed that the synthesized messages helped pilots to solve 16 out of 21 conflicts with a mean 5 s reaction time (see table below). In the other five cases, they have managed to solve the conflicts but declared that the messages strengthened their strategy. All the participants applied the expected procedure (reducing the selected speed, or reconnecting the autopilot). One participant (for whom the consistency checker failed to detect the conflict) took over the automation and manually managed the conflict.

conflict #	message helped out (out of 7)	mean RT (s)
1	5	6
2	5	2
3	6	7

The results of this technical demonstration are promising and show that generic solutions could be easily implemented in the cockpit. The consistency checker, based on the analysis of "automatic mode changes", allowed for the detection of different conflictual situations. Moreover the messages objectively helped pilots to solve conflicts and prevented them from experiencing "automation surprise": all participants managed to deal with the conflicts with short reaction time. The participants were unanimous in finding the messages to be more efficient than the current user interface. However, the interpretative value of these results remains limited as this proof-of-concept was conducted with few participants facing only a limited number of conflicts. Also we did not include a control group. There is a need to conduct a controlled experiment involving more participants and other "automation surprise" situations. Moreover, the design of the message has to be improved and tested as, for instance, most of our participants declared that the visual messages could be more efficient.

GENERAL CONCLUSION AND DISCUSSION

This research intended to characterize pilots' cognitive impairment during "automation surprise" situations that are known to jeopardize flight safety. The analysis of eye movement revealed that such conflicts impair attentional abilities, leading to an excessive visual search and in inability to extract relevant information. Interestingly enough, our results shown that the explore/exploit ratio, derived from the work of [18], was a simple but relevant way to indirectly measure the state of surprise and confusion. Thus, this ratio could be used to diagnose human operator states when facing a much wider variety of safety-critical situations, such as any technical breakdowns and related warnings that may surprise them. Moreover the use of this ratio could be extended to other domains (eg. driving) as its computation does not depend on areas of interest but on how the eye moves and processes information.To our knowledge, this study was the first attempt to objectively characterize pilots' performance and ocular behavior when they "*are surprised by the behavior of their strong but silent machine partners*" [34]. Corresponding to Sarter's intuitive reasoning, we therefore proposed to consider "silent" mode transitions (i.e. triggered automatically to self-regulate autopilot) as precursors of potential conflicts. Such an approach paved the way to the implementation of a real-time assistance system with minimum *a priori knowledge* contrary to most common approaches [8, 20, 30, 31]. Though we have no guarantee that the formal model detects all of the possible conflicts, the presented principle of detecting a hidden transition is applicable to any finite-state automata system commonly used in many critical systems (e.g. nuclear powerplant, submarine, train). The verification of completeness of this approach should be done by automation designers for each aircraft and validated with pilots during experimental campaigns. Moreover, our assistant system successfully synthesized appropriate messages and enabled the pilots to scrutinize the autopilot model. For sake of simplicity, we used auditory messages. However, this modality may be less efficient in noisy cockpit especially under high workload conditions [35]. An improvement would be to experiment with other designs combining auditory and visual messages displayed in the upper part of the primary flight display. Despite experimental limitations, our proof-of-concept shows that straightforward solutions should be implemented in today's flight deck and contribute to enhance flight safety. Since the remote eye tracking technology is available nowadays and its application in real flight conditions is possible [10], it would be interesting to evaluate the eye tracking for inferring pilot's attentional state and for confirming the occurrence of cognitive conflict in future flight deck. This additional inference would prevent from sending spurious messages (as occurred five times in our proof-of-concept testing when pilots managed to rapidly diagnose autopilot behavior). Moreover, this check could limit the effect of overreliance on automation [26] as it would prevent pilots from systematically relying on synthesized messages.

REFERENCES

1. Billings, C. E. *Aviation automation: The search for a human-centered approach*. Mahwah, NJ: Erlbaum, 1997.
2. Boeing. Statistical summary of commercial jet airplane accidents. worldwide operations 1959 -2011, 2012.
3. Breen, B. C. Controlled flight into terrain and the enhanced ground proximity warning system. *Aerospace and Electronic Systems Magazine, IEEE 14*, 1 (1999), 19–24.
4. Bureau Enquetes et Analyses. Final Report on the Accident on 1st June 2009 to the Airbus 1330-203 registered F-GZCP operated by air france flight AF447 Rio de Janeiro-Paris, 2012.
5. Butler, R. W., Miller, S. P., Potts, J. N., and Carreno, V. A. A formal methods approach to the analysis of mode confusion. In *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, vol. 1, IEEE (1998), C41–1.
6. Castelfranchi, C. Conflict ontology. In *Computational conflicts*. Springer, 2000, 21–40.
7. Chialastri, A. Automation in aviation, 2012.
8. Crow, J., Javaux, D., and Rushby, J. Models of mechanised methods that integrate human factors into automation design. In *International Conference on Human-Computer Interaction in Aeronautics: HCI-Aero 2000*, Citeseer (2000).
9. De Keyser, V., and Woods, D. Fixation errors: Failures to revise situation assessment in dynamic and risky systems. In *Systems reliability assessment*. Springer, 1990, 231–251.
10. Dehais, F., Causse, M., and Pastor, J. Embedded eye tracker in a real aircraft: new perspectives on pilot/aircraft interaction monitoring. In *Proceedings from The 3rd International Conference on Research in Air Transportation. Fairfax, USA: Federal Aviation Administration* (2008).
11. Dehais, F., Causse, M., and Tremblay, S. Mitigation of conflicts with automation use of cognitive countermeasures. *Human Factors: The Journal of the Human Factors and Ergonomics Society 53*, 5 (2011), 448–460.
12. Dehais, F., Causse, M., Vachon, F., and Tremblay, S. Cognitive conflict in human–automation interactions: a psychophysiological study. *Applied ergonomics 43*, 3 (2012), 588–595.
13. Dehais, F., Goudou, A., Lesire, C., and Tessier, C. Towards an anticipatory agent to help pilots. In *AAAI 2005 Fall Symposium" From Reactive to Anticipatory Cognitive Embodied Systems", Arlington, Virginia* (2005).
14. Dehais, F., Tessier, C., and Chaudron, L. Ghost: experimenting conflicts countermeasures in the pilot's activity. In *IJCAI*, vol. 3 (2003), 164–168.

15. Diez, M., Boehm-Davis, D. A., Holt, R. W., Pinney, M. E., Hansberger, J. T., and Schoppek, W. Tracking pilot interactions with flight management systems through eye movements. In *Proceedings of the 11th International Symposium on Aviation Psychology* (2001), 1–6.
16. Easterbrook, S. Handling conflict between domain descriptions with computer-supported negotiation. *Knowledge acquisition 3*, 3 (1991), 255–289.
17. Feary, M. Automatic detection of interaction vulnerabilities in an executable specification. In *Engineering Psychology and Cognitive Ergonomics*. Springer, 2007, 487–496.
18. Goldberg, J. H., and Kotval, X. P. Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics 24*, 6 (1999), 631–645.
19. Huettig, G., Anders, G., and Tautz, A. Mode awareness in a modern glass cockpit. In *Proceedings of the 10th International Symposium on Aviation Psychology*, vol. 1 (1999), 130–135.
20. Javaux, D. A method for predicting errors when interacting with finite state systems. how implicit learning shapes the user's knowledge of a system. *Reliability Engineering & System Safety 75*, 2 (2002), 147–165.
21. Leveson, N., Pinnel, L. D., Sandys, S. D., Koga, S., and Reese, J. D. Analyzing software specifications for mode confusion potential. In *Proceedings of a workshop on human error and system development, Glasgow, Scotland*, Citeseer (1997), 132–146.
22. Leveson, N. G., and Palmer, E. Designing automation to reduce operator errors. In *Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. IEEE International Conference on*, vol. 2, IEEE (1997), 1144–1150.
23. Mack, A., and Rock, I. *Inattentional blindness.* The MIT Press, 1998.
24. Manor, B. R., and Gordon, E. Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks. *Journal of neuroscience methods 128*, 1 (2003), 85–93.
25. Norman, D. A. The 'problem' with automation: inappropriate feedback and interaction, not'over-automation'. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences 327*, 1241 (1990), 585–593.
26. Parasuraman, R., and Riley, V. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society 39*, 2 (1997), 230–253.
27. Pizziol, S., Tessier, C., and Dehais, F. Petri net-based modelling of human–automation conflicts in aviation. *Ergonomics 57*, 3 (2014), 319–331.
28. Poole, A., and Ball, L. J. Eye tracking in HCI and usability research. *Encyclopedia of human computer interaction 1* (2006), 211–219.
29. Regis, N., Dehais, F., Tessier, C., and Gagnon, J.-F. Ocular metrics for detecting attentional tunnelling. *Human Factors and Ergonomics Society–Chapter Europe, Toulouse, France* (2012).
30. Rushby, J. Using model checking to help discover mode confusions and other automation surprises. *Reliability Engineering & System Safety 75*, 2 (2002), 167–177.
31. Rushby, J., Crow, J., and Palmer, E. An automated method to detect potential mode confusions. In *Digital Avionics Systems Conference, 1999. Proceedings. 18th*, vol. 1, IEEE (1999), 4–B.
32. Salthouse, T. A., and Ellis, C. L. Determinants of eye-fixation duration. *The American journal of psychology* (1980), 207–234.
33. Sarter, N. B., Mumaw, R. J., and Wickens, C. D. Pilots' monitoring strategies and performance on automated flight decks: An empirical study combining behavioral and eye-tracking data. *Human Factors: The Journal of the Human Factors and Ergonomics Society 49*, 3 (2007), 347–357.
34. Sarter, N. B., Woods, D. D., and Billings, C. E. Automation surprises. *Handbook of human factors and ergonomics 2* (1997), 1926–1943.
35. Scannella, S., Causse, M., Chauveau, N., Pastor, J., and Dehais, F. Effects of the audiovisual conflict on auditory early processes. *International Journal of Psychophysiology 89*, 1 (2013), 115–122.
36. Tintarev, N., Kutlak, R., Oren, N., Van Deemter, K., Green, M., Masthoff, J., and Vasconcelos, W. SAsSy - scrutable autonomous systems. *Proceedings of The Society for the Study of Artificial Intelligence and the Simulation of Behaviour 2013* (2013).
37. Tracy, J. I., Mohamed, F., Faro, S., Tiver, R., Pinus, A., Bloomer, C., Pyrros, A., and Harvan, J. The effect of autonomic arousal on attentional focus. *Neuroreport 11*, 18 (2000), 4037–4042.
38. Wiener, E. L. Human factors of advanced technology ("glass cockpit") transport aircraft. NASA contractor report no. 177528, Moffet Field, NASA-Ames Research Center, 1989.
39. Wiener, E. L., and Curry, R. E. Flight-deck automation: Promises and problems. *Ergonomics 23*, 10 (1980), 995–1011.
40. Woods, D. D., and Sarter, N. B. Learning from automation surprises and "going sour" accidents. *Cognitive engineering in the aviation domain* (2000), 327–353.