Flight procedures automation: Towards flight autonomy in manned aircraft

César Álvarez
Human Factors and Flight Deck
Airbus Defence and Space
Getafe, Spain
cesar.a.alvarez@airbus.com

Maria Isabel González
Avionics Architectures
Airbus Defence and Space
Getafe, Spain
maria.i.gonzalez@airbus.com

Alejandro Gracia
Avionics Architectures
Airbus Defence and Space
Getafe, Spain
alejandro.gracia@airbus.com

Abstract—Safe aviation operations rely on the flight crews’ adherence to operating procedures for normal, abnormal and emergency situations. Crew operations can be enhanced by applying a certain automation level to procedures (both on commanded actions and monitoring aspects), thus reducing pilot workload and increasing the crews’ mental spare capacity. Succeeding in partial automation of flight procedures is deemed a first step towards flight autonomy substantiated on Artificial- Intelligence (AI) based systems in the long term. It also paves the way towards reduced crew or unmanned operations.

This paper presents Cockpit Automation Procedures System (CAPS) as an initial prototype of this kind of automation. This technology, developed as contributor to Enhanced Flight Operations and Functions within the Clean Sky 2 framework, is capable of automatically executing crew procedures with an adjustable automation level and authority delegation and enables monitoring of pilot actions on cockpit controls.

A CAPS prototype has been developed considering representative operational scenarios to evaluate its contribution on crew workload and performance. An analysis is presented in this report that substantiates how the CAPS prototype design criteria have been established as well as automation levels. The analysis presented has targeted abnormal and emergency procedures such as engine failures, but the concept and prototype can be extended to other procedures.

Several findings have been uncovered during this exercise: first and foremost, the safety challenges that a potential certification of this critical function poses. Given its potential ability to control all aircraft, the function criticality, required design assurance, redundancy, override in case of failure, etc. are issues that need deeper analysis.

Other challenges include the human factors considerations such as loss of situational awareness depending on automation proposed or autonomy in the future. Besides, current aircraft systems and cockpit in the CAPS-hosting platform impose constraints that need to be addressed: CAPS could not be retrofitted on current aircraft cockpits, where systems’ controls are still ultimately based in mechanical or hydraulic means. Smart, adaptive cockpit controls with hybrid man-controlled and machine-controlled modes are required to enable CAPS and pilots coexistence.

This paper contributes in demonstrating the feasibility of including procedure automation into flight decks, while addressing the safety repercussions of the system actions, depicting the new Human Factors challenges arisen in the process and describing the next steps necessary towards full machine-controlled scenarios.

Keywords— procedures, automation, aircraft, cockpit, CAS, FWS, autonomy, Clean Sky

I. INTRODUCTION

Civil aviation development is nowadays progressing towards more autonomous aircraft. Many steps have been already taken and matured in this direction, like the implementation of autopilot functionalities, auto-land systems, auto-brake systems, etc. which despite currently needing pilot monitoring for a safety reason, they permit the aircraft to operate autonomously under some circumstances. The Cockpit Automated Procedures System (CAPS) concept intends to be another brick towards aircraft autonomous operations, yet in contrast to some of the mentioned systems, it focuses on cockpit procedure management and how the general systems are reconfigured. In order to allocate the CAPS in the context of crew task automation, Fig. 1, extracted from [1], depicts the different tasks that a flight crew performs along with systems that support or automate the task, including some of the ones mentioned before. The CAPS would contribute to Systems monitoring / management part.

II. STATEMENT OF THE PROBLEM

State of the art flight decks are operated by flight crews, adhering to operating procedures for normal, abnormal and emergency situations. This operational concept has been used throughout many years, proving to enhance safety level. Crew procedures generally differ in their execution depending on their type: whereas normal operating procedures are learnt and trained by flight crews, memory executed during operation and checked after their execution by means of checklists, abnormal and emergency procedures- although trained as well- are usually performed following read and do action lists, except for immediate actions, which due to their safety implications, need to be conducted by memory.

Procedures are generally paper based and/or displayed in cockpit displays. State of the art aircraft feature Electronic Centralized Aircraft Monitor (ECAM) or Engine-Indicating And Crew-Alerting System (EICAS) systems, which help flight crews in systems management and procedures and checklists execution. In terms of procedures execution support, these systems not only display the procedures in a dedicated display as well as system status and alerts, but already include some level of automation:
These systems trigger the procedures and checklist to be executed by linking them to raised alerts or specific logic conditions.

- They can sense some action when correctly performed by the crews.
- They can retrieve the status condition of some systems parameters in order to avoid the crew performing unnecessary steps.

However, ECAM/EICAS systems do not command nor request actions to systems so it is still the crew responsibility to correctly apply and verify the actions stated in the procedures, mainly referring to other cockpit controls in combination with ECAM/EICAS. The CAPS is intended to cover this gap by actuating directly into the systems from a unique interface with a varying degree of authority delegation in which the crew could or could not remain in the decision loop, eventually permitting autonomous system management.

1 Fully autonomous system management is out of the scope of the CAPS concept presented in this report.

III. ENVISAGED ROADMAP TOWARDS AUTONOMY

Autonomy in Procedures Management (as a pillar of flight autonomy in manned aircraft) is pictured as a long-term goal with several steps as shown in Fig. 2.

1) Deterministic procedures automatization for bound use cases (CAPS). This stage is based on data capture and procedures automatization, based on deterministic rules and pre-configured delegation rules. This is CAPS current scope and the focus of this paper.

2) Decision making algorithms with pilot control: based on a mix of deterministic and non-deterministic (machine-learning-based) automatization. Full delegation is not granted to the automating system at this stage due to safety constraints and concerns. Reduced flying crews (semi-Single Pilot Operation) could be envisaged.

3) Decision making algorithms with pilot monitoring: In this stage the role of the pilot as responsible of the “aviate” function is notably reduced, operating as flight monitor and taking action as a fallback actor.

4) Full autonomy

IV. ACTION TYPES

Focusing on the first stage of the aforementioned roadmap, the automation mechanism depends on the crew action types.

If an emergency procedure is used as an example, a number of different action steps can be identified. There are some steps that require the flight crew to act on a cockpit control (e.g. shut down fuel feeding valve to an engine, discharge a fire extinguishing agent), decide the next step to take depending on a condition (e.g. if engine fire persist do certain things), focus on other task such as communications (e.g. communicate emergency to ATC) or perform strategic decision (e.g. land at the nearest suitable airport).

CAPS focuses only in those steps that require the crew to act on a cockpit control, leaving others out of the scope. Several levels of automation have been established from a manual action to an autonomously performed action with general criteria for their use (Fig. 3):
Manual actions would mainly correspond to actions that have a physical limitation to be automated (e.g. put on face mask) or flying tasks requiring Hands On Throttle And Stick (HOTAS), that are left for pilot manual command as a general philosophy.

Pilot confirmed actions correspond to those actions that have some inherent irreversibility or critical safety repercussions, and thus need pilot authorization prior to their execution.

Automatic actions are those that do not fall in any of the above mentioned groups and therefore subject to be automated.

As a design philosophy, usually automatically executed actions are nested by a higher level clause by which the pilot can delegate the authority to the system. For instance, securing the engine in case of a fire requires multiple switches actuation which can be automatically done and nested under a higher level clause that the pilot needs to confirm. This can be applicable to a whole procedure, should it be the case. Further details on the level of automation are discussed later in this paper.

Beyond these actions that can be deterministically automated, the “aviate”, “communicate” or strategic decisions cannot be scoped with the current certifiable technologies given their safety implications.

These strategic decisions often require a great number of high-variability sub-tasks to be completed. For example, in order to perform a “Land at the nearest suitable airport” action, the crew will need to decide which airport and approach path to follow, based on a number of inputs from a wide variety of sources: aircraft status, airports in the vicinity and their constraints (visibility and weather, Estimated Time of Arrival, applicable Notice to Airmen (NOTAMs)), crew and passengers condition, etc. Whereas an algorithm based on deterministic logics may be complex to develop for situations like these, Machine Learning (ML) and Deep Learning (DL) algorithms would fit better herein.

The implementation of these ML-based algorithms can in fact respond to several levels of user interaction. On one side, the system may propose one or more options for the strategic decision for the crew to confirm and execute or could autonomously decide and execute without crew intervention. This latter step would pave the way to fully decision making process.

The CAPS has been conceived as a system to automate flight deck procedures with main focus on abnormal and emergency procedures. As for system design, CAPS does not handle the warning management function (i.e., it will not replace the aircraft Warning System), but it will be slaved to the existing aircraft Warning System, capturing the Flight Warning System (FWS) alerts as triggering conditions for launching abnormal and emergency procedures automatization.

The CAPS assumes the existence of an internal database of procedural actions linked to alert triggers, each of them tagged with a delegation or automation level. These levels are (adapted from [2] and [3]):

- **Level 0**: The action is purely manual and the crew needs to execute it manually and notify the system it is completed. The CAPS does not intervene.
- **Level 1**: The action is purely manual and CAPS can sense the status, so that when the system status corresponds to the required status, the FWS senses the alerting conditions are not active anymore (and thus resets the alerting state).
- **Level 2**: The action is performed by CAPS, but requires confirmation by the flight crew prior to its execution.
- **Level 3**: The action is performed by CAPS without confirmation from the flight crew.

whereas the following criteria are used for allocating each task to each level depending on its nature:

- HOTAS actions normally operated by the Pilot Flying, flight limitations, Communications actions or those that are not feasible for practical reasons will be Level 0 or Level 1.
- Actions over cockpit controls other than HOTAS will be automatic without pilot intervention (i.e. Level 3).
- When a group of automatic actions are to be executed they will be nested by a higher-level Level 2 action.
CAPS is intended to intercede between the aircraft systems and FWS. This kind of actions will include description on what the system will do for the crew to crosscheck before confirming.

When an abnormal or emergency condition is detected by the FWS, the FWS notifies the CAPS of an existing alerting condition, which, according to the CAPS database, requires a series of system state changes to be performed. CAPS will provide visual indications to the crew of the expected actions (the procedure steps) and their automation level, as a sort of electronic checklist. If responsibility for actions execution is delegated to CAPS, it will perform the expected procedural actions by commanding system controls with or without crew confirmation before CAPS executes them, depending on the level of the alert (otherwise they will be carried out manually by the flight crews).

Although the main focus and principal benefits of CAPS automation are found during the automation of abnormal and emergency procedures, non-failure related triggers can also be detected by CAPS in order to execute normal procedures, either manually triggered by the crew or automatically triggered by any aircraft condition logic (e.g. aircraft position, time, flight phase change, etc.).

In order for the CAPS to properly work, an initial adaptation of the procedure “as is” is required. Extra higher-level actions are added so that they permit grouping lower-level tasks, either at a whole procedure level or as a subset of actions. This would permit the automation of a series of lower level actions. Besides, procedure actions can be reordered when performing them in sequence is not necessary and it helps action grouping. To exemplify this, the following example is provided, based on a generic Engine Fire procedure (Fig. 7).

The procedure “as is” includes notifying ATC of the emergency, prior to checking fire persistence and discharging the second fire extinguishing agent if persists, which requires a minimum actuation time. For the CAPS, fire extinguishing actions can be grouped together and separated from the communications task. Besides, power lever actuation is necessary at the beginning of the procedure, which based on the HOTAS philosophy needs to be segregated as well from the former.

The Power lever actuation will be a Level 1 action type, since the position can be sensed, whereas ATC notification will correspond to Level 0 as no means for sensing can be easily implemented. The rest of actions can be automated as Level 3 actions; however they need to be nested to higher Level 2 action, for instance “ACTIVATE L ENG FIRE EXTINGUISHING?”, that upon crew acceptance will unleash the automatic execution of the nested lower level actions. In
this case, a warning note may be added to notify the crew that confirmation of such action would trigger irreversible actions like discharging extinguishing agent. This will maintain crew crosscheck and an equivalent confirmation level on the actions. The interface concept could be as shown below in Fig. 8.

The remaining actions following the automated procedure completion will consist of reviewing a summary page on the actions being taken, limitations, etc.; hence being similar to the current operation.

B. Methods

A prototype CAPS has been developed by Airbus Defence and Space for validation of this first stage of procedures automatization, integrated within a virtual aircraft (a simulated civil Airbus A330 aircraft) and an “Active Cockpit” embedding virtual panels that emulate the A330 real cockpit, which allows modifications for rapid prototyping of new crew interaction means for CAPS evaluation purposes.

The CAPS concept validation consists of a series of crew-in-the-loop evaluations in a set of flying scenarios that include a sample of system failures and associated procedures. Crew’s feedback on automation as well as perceived workload and situational awareness will be gathered for comparison with current operation.

These “as-is” procedures (in paper format) have been processed and adapted, following the criteria of the previous section, in a set of instructions (procedural actions) that conform to the CAPS database. Each action contains: the necessary data for providing visual indication of the automation level with expected crew response, and the specific action to be performed (which cockpit control needs to be acted-on and/or monitored). This database has been created using a textual markup language in order to allow nesting and easy modification if required.

During the simulated scenario, the induced system failures will trigger the simulator FWS to raise alerts. These alerts are reported to the flight crew through the Master Caution and Master Warning cockpit lights, and visualized through an alerting text in the cockpit ECAM display with the appropriate legend and color to indicate its required awareness level.

The prototype CAPS application, hosted in a laboratory PC, monitors continuously the aircraft simulated buses, including the FWS output buses carrying this alert signal. CAPS parses the alerting text and queries its own database to recall the expected procedural actions linked to it.

For each procedural action, CAPS provides the ECAM display with an electronic procedure containing the set of visual indications for each expected action, including specific coloring, indentation and checked/unchecked boxes depending on the action type, nesting level and completion status. ECAM display has been selected as the most adequate location for the indications of CAPS procedural actions as it presents failures and proposed corrective actions in a unified manner (Fig. 8 and Fig. 9).

The automation process starts at this point, with CAPS carrying out an expected behavior for completing each procedural action as pre-programmed in the database:
For Level 0 Actions, CAPS stands by waiting for a manual confirmation by the crew that the action has been carried out. For this purpose, an additional crew control was added to the cockpit virtual panels whose activation status is fed to CAPS.

For Level 1 Actions, the database indicates which cockpit controls are expected to vary upon crew action; CAPS monitors them, and when the expected status is sensed the procedural action is automatically considered complete.

For Level 2 Actions, CAPS stands by waiting for a delegation confirmation by the flight crew for execution, and when authorized, it commands the cockpit controls coded in the database with their expected setting as per the procedure. CAPS also senses automatically the completion of the performed actions and informs the crew accordingly.

For Level 3 Actions, CAPS acts similarly to level 2 actions, but without asking the crew for execution permission.

In all cases, whenever a procedural action is complete, CAPS updates the Human Machine Interface (HMI) to indicate the crew the action execution status.

In this approach, CAPS acts as a sort of “parallel user” of the cockpit controls, with a full vision of the cockpit status, and with the possibility of acting on any cockpit control regardless of crew actions. This of course implies having a set of cockpit controls that allow being controlled by an external actor (the crew) and an internal actor (CAPS), which is possible for the prototype by using virtual electronic panels, but it is not the case in a real, current A330 aircraft.

VI. ISSUES UNCOVERED / CHALLENGES FOUND / DISCOVERIES

A. Safety and certification issues

CAPS prototype implementation is aimed to validate the functional and operational sides of the procedures automation, in the simplified environment that the simulated aircraft provides, where redundancy and a fully representative set of interfaces and systems is not implemented. However, should the CAPS concept be put in place in a real aircraft, the CAPS system design process would need to address its impact on the existing aircraft systems architecture.

Usually, in the design of Integrated Control Panels (ICPs) and in order to mitigate the more demanding safety consequences of erroneous or unintended commands to the aircraft systems, the commands are independently sent from simple mechanical devices by different communication means (e.g. discretes + redundant CAN bus) with electrical independence as to avoid a single source of failure. The integration of a system like CAPS would introduce a complex software variable in the game which would require the pertinent safety analysis for ensuring a proper deployment of the necessary development assurance processes.

Integrating a new complex agent in the aircraft, potentially with the same authority than the pilots, implies a full reassessment of all the functional threads where pilots have any action responsibility, either for correct operation or to apply failure mitigation actions. Taking into account that a failure of cockpit controls can have catastrophic consequences, the first foreseeable issue is that the required development assurance level of CAPS would be the highest one (DAL A as per [5] and [6]).

For the first automation stage of the roadmap where CAPS is intended to operate in collaboration with a flying crew, pilots’ intervention in case of CAPS suspected failure can relax the robustness requirements imposed on the system (as pilots retain full authority). This can be the case still in further steps where machine learning algorithms are foreseen to complement the pure deterministic behavior of CAPS. However, for the last step where full autonomy is desired, it remains unclear the risk mitigation techniques to be applied in terms of how to manage having a top supervisor of the automation process.

The certification of equipment in large aeroplanes poses one of the main entry barriers for the use of systems whose aim is to enhance the autonomy capabilities of the aircraft. The Certification Specifications from European Aviation Safety Agency (EASA) (that could be generally applicable also to other Certification Authorities) impose stringent requirements [9] such as CS-25.1301 (Function an Installation), CS-25.1302 (Installed Systems and equipment for use by the flight crew) and CS-25.1309 (Equipment, systems and installations).

These are very general and broad requirements which act as a trap net from which no system to be installed in an aircraft can escape, whose major aim is to show to the Certification Authorities that the system performs as intended in all foreseeable situations.

It is only recently that EASA has published an AI roadmap [7] which consists basically in three incremental steps towards full autonomy, as depicted in Fig. 10.

The CAPS described in this paper falls within the first milestone of “Crew Assistance / Augmentation” of the EASA’s AI certification roadmap.

The major certification challenges for AI, as described by EASA, reside on:

- Traditional development assurance frameworks are not adapted to machine learning
- Difficulties in keeping a comprehensive description of the intended function
- Lack of predictability and explainability of the ML application behavior
- Lack of guarantee of robustness and of no ‘unintended function’
- Lack of standardised methods for evaluation of the operational performance of the ML/DL applications
- Issue of bias and variance in ML applications
- Complexity of architectures and algorithms
- Adaptive learning processes
Just in March 2020, EASA has deepened into the Learning Assurance concept as a parallelism to the traditional Development Assurance methodology [8]. This concept could serve as an additional Acceptable Means of Compliance (AMC) towards achieving the demonstration of those general system requirements, when AI algorithms are involved. A “W”-shaped Learning Assurance life-cycle is proposed (see Fig. 11) in contraposition to the classic “V-shaped” development life-cycle, where emphasis is put into the verification of the learning process and AI model training, so the intended function of the system can be guaranteed within specific boundaries. The Data life-cycle management of the model now becomes a center piece of the whole development process and is the key factor to ensure the intended function of the system.

Another highlighted technique is the Statistical Learning Theory (SLT) applied for guaranteeing bounds in the generalization gap, or in other words, having a finite training sample that could enable accurate unseen predicted data. This could mitigate the uncertainty and lack of predictability of the ML behavior during operation.

B. Human Factors challenges

The implementation of the CAPS presents a number of Human Factor challenges, which need to be carefully addressed.

First, the “workload/situational awareness/error” trade-off needs to be assessed. On one side the workload reduction provided by automation is clear, as the process of reading the action on a procedure, finding the associated control, actuating on the control and verifying the systems status change is done autonomously after the authority is delegated to the system (Level 2 and Level 3 actions). Same happens to the possibility of errors due to the actuation on wrong controls (to the extent allowed by the CAPS robustness, ideally improving pilot’s performance). However, CAPS automation can have effects on the situational awareness for the flight crew, since the systems are automatically configured without their intervention. In the Airbus current operational concept, after a procedure has been completed, the flight crew reviews the STATUS page in the ECAM, which provides a summary of the aircraft status, including inoperative systems, flight limitations or any other relevant information for the flight crew. The review of this information becomes more relevant with the CAPS concept so that the crew is aware of the overall aircraft status, in case strategic decisions need to be taken throughout the rest of the flight based on such status. Stress should be put in this part, through training areas or even by system design, reinforcing the means by which this information is provided to the crew.

The aircraft status review is key to avoid ending up in clumsy automation as defined in [4], and making a more complex flight strategic decision.

Another concern is the reliance of the crew in such a system, due to its level of novelty. The CAPS is targeting abnormal and emergency situations, in which the safety of the aircraft is compromised and therefore the crew level of stress can be moderate. In order to compensate reliance in the system, alleviating means have been initially provided in the CAPS concept: permitting the crew to visualize what the system is doing and permitting the crew to interrupt the automation at any point. The visualization of current CAPS operation is not required per the automatic procedure concept, since the crew will not need to see “what the system is doing”, but “what the system can and will do” and “what the system has done”; however, feedback from operators suggested this approach.

C. Systems impact: Architecture and implementation

The impact of CAPS integration with a state-of-art cockpit and real systems would require a reformulation on the cockpit philosophy, which would result in new cockpit designs.

Current cockpits are operated by the crew only, however in a CAPS integrated configuration CAPS itself is another system that can actuate systems’ controls. Besides, existing cockpit controls design embeds, by majority, mechanical or hydraulic action mechanisms (bi-stable switches, physical toggles, position-stable rotary controls), with a deliberately simple design and with the capacity of providing a haptic feedback to the pilots. A means to provide control access, haptic feedback (for the crew) and consistent control status information regardless of the actuation agent (crew or CAPS) is a must in this hybrid environment.

Several means can be used to address the CAPS needs to interact with a large number of cockpit controls that are dual-user capable:

- Reducing the number of physical controls in the cockpit, so that system control is implemented by virtual controls integrated in displays located in the main instrument or overhead panels.
- Maintaining physical controls and using self-actuation means to physically modify their position.
Maintaining physical controls, but using momentary switches with lighted status indication and display instead of bi-stable switches and toggles and infinite-stepped rotary wheels with position indication instead of position-stable rotary controls.

In any of the mentioned cases, it would be possible to simultaneously operate a system manually or by means of automation without leading to position-status inconsistencies that would induce crew error. This issue is similar to auto-throttle or auto-pilot systems with respect to flight and thrust controls.

On top of that, in a CAPS integrated environment, the CAPS needs to interface with a large amount of signals monitored/controlled from the cockpit. In some modern aircraft, the cockpit controls have been integrated in centralized smart cockpit panel systems (e.g. ICPS in A350, A380, and A400M) with health management and integrated data acquisition and distribution capabilities. In legacy aircraft platforms though, cockpit controls mainly rely on discrete and analogic interfaces. Since the market tendency is to avoid using computers with too many interfaces (in favor of extending the use of several distributed data concentrators as part of a network controlled by a few computing resources), the most likely configuration for a CAPS system is to build upon a distributed Integrated Modular Avionics (IMA) architecture.

Thinking ahead about the implementation of machine learning algorithms to complement or even substitute the CAPS deterministic processing in future phases, the aviation industry in general faces the challenge of how to allow non-deterministic systems to carry out critical functions [7].

Last but not least, the robustness of the new system (required to comply with the certification and safety constraints) will imply, not only demonstrating high design assurance development processes, but also incorporating backup or override modes to allow pilots intervention in case of failures. Current cockpit controls use simple hardware design to minimize complexity as much as possible or to provide dissimilar backup means – CAPS hybrid cockpit controls will need to keep these simple designs at least for the most critical cases, at least until enough in-service experience exists to demonstrate that the system alone can automate procedures with a minimum to none human mediation.

**VII. CONCLUSIONS**

Some conclusions can be derived from the concept presented in this document. First, the implementation of a system that can automate cockpit procedures is feasible, when limited to actions that involve actuation over controls. This implementation implies a number of cockpit and operational changes that need to be addressed while ensuring compliance with safety and certification aspects. Second, the automation of complex strategic actions would be in the scope of machine learning algorithms. Finally, the concurrent deployment of automated procedures, based on current deterministic schemes and on machine learning algorithms is necessary in order to progress towards safe civil aviation autonomous operations.

**ACKNOWLEDGMENT**

This project has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 945583. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union.

![Clean Sky Logo]

Clean Sky is the largest European research program developing innovative, cutting-edge technology aimed at reducing CO2, gas emissions and noise levels produced by aircraft. Funded by the EU’s Horizon 2020 program, Clean Sky contributes to strengthening European aero-industry collaboration, global leadership and competitiveness.

**DISCLAIMER**

The results, opinions, conclusions, etc. presented in this work are those of the author(s) only and do not necessarily represent the position of the JU; the JU is not responsible for any use made of the information contained herein.

**REFERENCES**

[1] Marie Lacabanne, Franck Armadieu, André Tricot, Nathalie Spathgier-Gaillard. “Analysis of the flight task around different types of aircraft”. International Conference on Human-Computer Interaction in Aerospace (HCI-Aero), Sep 2012, Bruxelles, Belgium. hal-01627366

[2] Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). “A model for types and levels of human interaction with automation”. IEEE Transactions on systems, man, and cybernetics—Part A: systems and humans, 30(3), 286-297.

[3] Sheridan, T. B. (2000). “Function allocation: algorithm, alchemy or apostasy?” International Journal of Human-Computer Studies(52), 203-216.

[4] Lee, John & Seppelt, Bobbie. (2009). “Human Factors in Automation Design”. 10.1007/978-3-540-78831-7_25.

[5] RTCA DO-178B “Software Considerations in Airborne System and Equipment Certification”. (Dec 1992)

[6] RTCA DO-254 “Design Assurance Guidance for Airborne Electronic Hardware”. (April 2000)

[7] “Artificial Intelligence Roadmap, A human-centric approach to AI in aviation”, EASA, Feb. 2020

[8] “Concepts of Design Assurance for Neural Networks (CoDANN)”, EASA, March. 2020

[9] “CS-25 Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes”, EASA