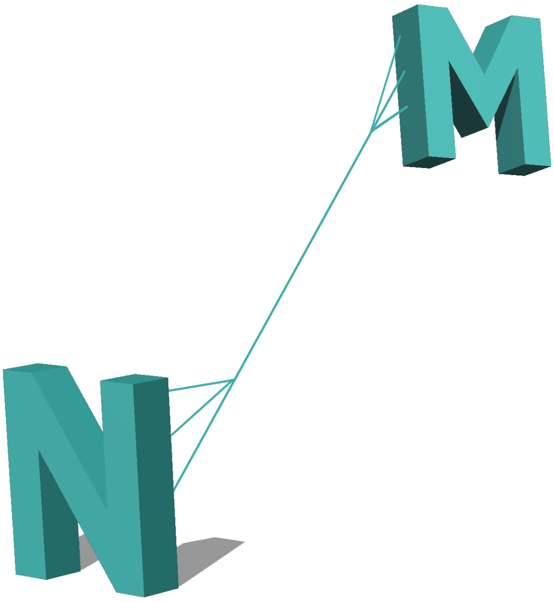
NtoM

CONOPS FOR A SAFE INTEGRATION OF MULTI-RPAS OPERATIONS IN CIVIL AIRSPACE

MIGUEL ÁNGEL FAS MILLÁN



Advisor: Dr. Enric Pastor Llorens

Thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy, as part of the
PHD PROGRAMME IN COMPUTER ARCHITECTURE

September 2019

Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

CONTACT
☞ mfas@ac.upc.edu

orcid.org/0000-0001-8849-2799

ABSTRACT

The gradual integration of remotely piloted aircraft systems (RPAS) in civil airspace, sharing airways with commercial flights, is expected to be completed in a few years once the legal issues and those regarding the unmanned traffic management are solved. This will open the floodgates to a myriad of new services and a demand that will probably face a lack of pilots, something that can already be found in the manned case.

The concept of operations proposed in the thesis pursues the feasibility, from a human factors perspective, of having a single pilot/aircrew controlling several RPAS concurrently in non-segregated airspace. To achieve such feasibility, this multitasking should be safe, and not interfere with the job of the air traffic controllers due to delays or errors associated to the parallel piloting. To this extent, a set of tools and measures at several levels are suggested, which includes workload balance and prediction, action monitoring and interface usability.

The management of the workload includes the consideration of the personalised workload profile of the pilots to determine their workload threshold and time requirements while executing the tasks. Based on these profiles, the allocation of resources will try to allow the parallelisation of tasks and therefore the increase of the productivity while providing a safe margin of total workload and flexibility in the strategies chosen for the management of the tasks. The balance of the workload requires the prediction of it, which would be based on the aggregation of different sources of information: aircraft readings, scheduled tasks, external reports and a suggested map of wokload patterns usually found in the areas overflown.

The monitorings are justified by the fact that the concurrent piloting could be misleading, so the system provides a safety check, acting as a kind of first officer.

The reason behind the focus put on usability is that only a quick interface providing an appropriate awareness could allow a safe and productive parallelisation of tasks.

While the airline is the main beneficiary of a concept like this, because the productivity of its human resources and the availability of the service would be multiplied, it also offers benefits for a one-pilot-one-aircraft ratio, as it provides extra safety measures or the possiblity to parallelise other roles in the flightcrew.

To illustrate some of the features of the proposal and test them with pseudo-pilots, a prototype was implemented and some experiments conducted to compare the performance with or without some of the measures implemented. These showed a decrease in the number of errors, oversights, and subjective stress, and were useful to inspire improvements and new features.

Some of the suggested elements of the concept could also be leveraged outside of it. For instance, it relies greatly on the exploitation of the potential of Controler-Pilot Data Link Communications (CPDLC), anticipating a future widespread implementation and full use. A CPDLC display for RPAS pilots was implemented focused on reducing the head-down of current implementations while providing a more descriptive status of the communications, both key aspects for the quick response that could be required from multi-RPAS pilots. This display, as a standalone application, could be used by pilots and controllers to learn an train the use of the CPDLC phraseology and composition rules. Also, the connectivity framework used for the communication between server and clients allows the simulation of different scenarios of Quality of Service of the link, which can be used to train the procedures to follow when related problems arise. Finally, the implementation of the suggested map of workload could also serve to air traffic management analysts to get the perspective of the pilots during the operations and detect hot spots.

The safety measures and monitorings were positively evaluated by the pilots and controllers surveyed even when some of them constituted redundant checks to existing air traffic control monitoring tools. The CPDLC display was quite well considered; also to be used by controllers, who found it very intuitive, quick and with the information clearly displayed and at hand. Finally, the procedure of handover suggested was well evaluated to avoid the errors and problems found during the delicate process of control migration.

Keywords: multi-RPAS, CPDLC, unmanned

CONTENTS

GLOSSARY

ADS-B Automatic Dependent Surveillance-Broadcast. Historically, the way the position of an aircraft was determined was by using a radar. With ADS-B, is the aircraft which periodically reports its position, altitude and speed. Anyone with an ADS-B receiver can read those data. 18, 38, 47, 99

ADS-C Automatic Dependent Surveillance-Contract. In ADS-B we had the aircraft periodically sending information with no previous request and open to anyone with a receiver. ADS-C is addressed only to the ATC, who sets the periodicity and data contained in the reports. 47, 48

API Application Programming Interface. Set of programming resources provided by a software component for other components to be able to use it. 16, 77

ATC Air Traffic Control. Service provided to organise and optimize the air traffic, prevent accidents and assist pilots. 2, 8, 14–16, 20, 23, 24, 26, 29, 30, 34, 38–41, 43, 46–48, 50–56, 58, 60, 62–68, 72, 74–76, 95, 97, 99, 101

ATM Air Traffic Management. 3, 5, 18, 23, 26, 57

ATS Air Traffic Services. 46

ATSU Air Traffic Services Unit. Defined by ICAO as "A generic term meaning variously, air traffic control unit, flight information centre or air traffic services reporting office" . 8, 15, 35, 46, 53

AVO Air Vehicle Operator. The pilot in charge of the aircraft, the equivalent of the Captain in manned aviation. 4–6, 77

C2 Command and Control. In this context, referred to the data link that allows the pilot to control the RPAS from the GCS. 35, 99

C3 Command, Control and Communications. When the C2 link is used also to transfer communications. 16, 18, 35, 59, 61

CDA Current Data Authority. Defined by ICAO as "The designated ground system through which a CPDLC dialogue between a pilot and a controller currently responsible for the flight is permitted to take place." . 35, 46

CDU Control Display Unit. In the current context refers to the device provided with a display and buttons used in the cockpit by the pilot as interface with the FMS. 39, 42, 100

ConOps Concept of Operations. Document describing a system: its purpose, structure, user roles, policies or procedures involved. Specifies its needs, limitations and could also guide a methodology of development. v, 1–6, 8–11, 17, 20, 28, 32, 33, 47, 57, 67, 73, 75, 76, 79, 95

CPDLC Controller-Pilot Data Link Communications. Seeks to base a great part of the controller-pilot communications in text messages sent through data link instead of using only voice. 8–10, 14, 18, 23, 24, 30, 34, 35, 37, 39–41, 44, 46–48, 50–53, 57, 63, 66, 68, 70–73, 76, 80, 91, 93–95

DDS Data Distribution Service. Connectivity framework standard for interoperable data exchange based in a publisher-subscriber schema. 18

eDEP Early Demonstration & Evaluation Platform Project. An Eurocontrol flight simulator. 100

FMS Flight Management System. Onboard computer able to carry out with many of the tasks of the pilot. Its cockpit interface is called Control Display Unit which, when it allows also data link communications, is called Datalink Control and Display Unit (DCDU). 8, 39, 41, 48, 60, 99, 100

FO First Officer. Second in command, this pilot assists the Captain and assumes that role in case of incapacitation. 4, 5, 20, 54, 55

GCS Ground Control Station. Fixed or portable facilities to control an unmanned vehicle. 3, 4, 7, 10, 13–16, 18, 21, 23, 25–28, 30, 32–35, 38, 39, 47, 48, 51, 53, 54, 57, 58, 61, 71, 72, 76, 77, 79, 80, 99, 100

GO Ground Operator. Member of the ground crew, supporting the aircraft from ground facilities. 5

ICAO International Civil Aviation Organization. 40, 46, 51, 73, 99

LLP Lost Link Procedure. Pre-loaded set of instructions that the RPA follows when the link with the remote pilot is lost and the Lost Link Timeout expires. 47, 48, 54, 58, 60–65, 67, 70, 72, 95, 100

LLT Lost Link Timeout. Period of time in lost link after which the FMS loads the LLP. 38, 60

LOA Level of Automation. In this context, the levels of automation would define configuration modes where some of the tasks are automated. In some cases, the operator could swap from one mode to another during the flight. 27

MCDU Multi-Function Control and Display Unit. While a CDU was usually dedicated to a single function, a MCDU is able to serve as an interface for different purposes. 39–41, 48

NPI NtoM Pilot Interface. The client application connecting the GCS to the NtoM system and providing the specific NtoM commands and communications. 7, 16, 18, 33–35, 48, 52, 57, 58, 71, 72, 74, 77

QoS Quality of Service. Here, refers to the level of performance of the data traffic, with a clear impact in the control and awareness of the pilot. 18, 23, 35, 59, 60

RAISE Extension of the eDEP simulator developed by ICARUS research group addressed to the modelling of RPAS operations. 18

RPA Remotely Piloted Aircraft. Unmanned aircraft. While it could be autonomous to some extent, it is assumed that there is always a human able to take control, unless there is a problem avoiding it. 2, 3, 7, 10, 16, 18, 19, 23, 24, 33, 34, 47, 57, 100, 101

RPAS Remotely Piloted Aircraft System. Unmanned aircraft and all the supporting equipment providing its control, like the ground control station or the data link. v, 1, 3–6, 9, 10, 16, 18, 24, 30, 33, 34, 39, 40, 47, 50, 51, 58, 59, 63, 64, 72–74, 77, 79, 99–101, 103

SES Single European Sky. Initiative of the European Commission to create an unified legislative framework for European aviation. v, 101

SESAR Single European Sky ATM Research. European Union's project to adjust the Air Traffic Management to fit the SES airspace redesign. v

SPO Single Pilot Operations. Piloting of an aircraft by a unique pilot. 3–5, 20, 55

UAS Unmanned Aerial System. Equivalent definition to that of RPAS but including autonomous aircraft that do not require a remote pilot. 16

UAV Unmanned Aerial Vehicle. Unmanned aircraft. This term embraces any level of autonomy; it can be fully autonomous, partially autonomous or being an RPA. 4, 18, 22, 32, 66

W/U WILCO/UNABLE. A message element whether confirming (WILCO) the execution of an instruction from the ATC or communicating the impossibility to accomplish it (UNABLE). 43, 49

1 INTRODUCTION

1.1 CONTEXT

The current mosaic of the European Union (EU) airspace, its flight paths and its national level boundaries and legislation, suppose an encumbrance to take advantage of the efficiency that nowadays navigation technologies provide to modern aircraft. To solve this, it was created the Single European Sky (SES)[1] project. The goal of this ongoing plan is to reduce the fragmentation of the EU airspace moving to a European level legislative framework, dividing it into Functional Airspace Blocks (FAB)[2] based on operational requirements rather than in national boundaries. Besides, a redraw of the flight paths will lead to save fuel, and reduce emissions, noise, delays and costs. One fundamental part of SES is the design and harmonization of the future architecture of the Air Traffic Management (ATM) systems, improved both from the technological and the operational point of view to increase safety, capacity and efficiency. This is the task of the Single European Sky ATM Research (SESAR)[3] project. Considering this process of evolution, any new project related to aviation should take into account this future scenario and being able to integrate the requirements defined in SES. That is the case of the integration of the Remotely-Piloted Aircraft Systems (RPAS).

The International Civil Aviation Organization (ICAO) is the agency of the United Nations (UN) that develops and publishes standards, procedures and protocols for the development of international air transport that are later adopted by the UN member states. One of the principles behind these guidelines is to guarantee equality in access to the airspace. This was the reason why in 2011 the agency set the basis for the development and integration of RPAS. As a response to that recommendation, in 2012, the European Commission (EC) created the European RPAS Steering Group (ERSG), formed by stakeholders in the area. This group developed and published in 2013 the European RPAS Roadmap[4], a work plan defining the steps, responsibilities and milestones to perform the gradual insertion of RPAS in the future SESAR operational concept.

At the time of writing this thesis, this progressive integration is already raising a whole new industry and a wide service sector and supply chain [5], with lots of new ideas of applications arising at an impressive pace, as usually seen when a new technology with a great potential appears. By 2050, the AeroSpace and Defence Industries Association of Europe (ASD) estimates about 150,000 jobs considering only those directly related to RPAS activities [6]. The civil usage on its own has an envisioned potential annual value of more than EUR 10 billion in the European market by 2035 [5]. But, above all, RPAS is bringing a broad spectrum of applications in many sectors that will provide many benefits to society.

The military have been quite some time exploiting the use of the RPAS, serving as a test bed for the current adoption at a civil level, similar as happened with other previous technological developments. Thus, the present work leverages the previous research focused on military operations, the reports gathering lessons and the resulting conclusions and guidelines from all that experience.

So far, civil RPAS are not equally sharing the airspace with the manned aircraft. Current civil applications are restricted and generally not included among commercial manned aircraft airways, except for the case where such inclusion is being tested. Frequently, they are integrated in specific areas, which grow in number at a good rate [7]. Although the following affirmation will hopefully be soon obsolete, the main bottleneck nowadays for the RPAS integration or inclusion in non-segregated airspace is not placed in the capabilities of the technology, but in the fragmented or undefined regulatory environment or the creation of an Unmanned Aircraft System Traffic Management (UTM), this last being addressed by several initiatives [8][9][10][11][12].

1.2 MOTIVATION AND OBJECTIVES

As it can be checked nowadays, the potential and flexibility of the unmanned aircraft opens up the market to a wide range of commercial services. This market, similar to any other business concept, will depend on sufficient profitability to survive and grow and any tool to support its safety or improve its productivity will mean an incentive to those investors required to give a boost to this emerging industry. In this economic sense, one way to optimise resources that the RPAS allow is to have one pilot or aircrew controlling several flights at once. This concurrent piloting can be safe enough if correctly addressed, and considerable research has been done in this direction [13][14][15]. The high Level of Automation (LOA) of these aircraft makes the multi-RPAS operations feasible [16], even considering single-piloted RPAS in the more challenging military environment [14][17]. RPAs are highly automated and to some extent autonomous by necessity. Detect-and-avoid systems are in charge of applying urgent actions that cannot allow delays in the control derived from the fact of being remotely piloted; these can also respond to events that can not be detected by the pilot due to a lack of awareness. If the RPA loses the link with the Ground Control Station (GCS) it should be able to execute some procedure to avoid damages, recover the aircraft, the link, or try to complete the mission by itself. Relying on a high automation and autonomy, the role of the pilot becomes more high-level, focusing on guiding and supervising the mission, instead of the old-style piloting. However, depending on the situation, such technological breakthrough shows its drawbacks. In some kinds of mission the pilot is constantly making decisions, or coordinating with the mission commander or the payload operators, but others contain long hours of monotonous supervision. A bored pilot is prone to a dangerous decrease in awareness and an increase in both reaction time and number of lapses and errors [18][19][20]. To avoid such disengagement, some solutions, like adaptive systems, automate or not some tasks depending on the current workload of the pilot. NtoM suggests a different approach: predicting the workload and assigning the flights to the pilots with a balance of workload that allows them to control more than one aircraft safely and comfortably. The system would overlap periods of activity and periods of monitoring of different flights to avoid boredom but also dangerous situations of excessive workload. Even when all flights were just monitored, the fact of having different flights, with their respective future or pending tasks and status, induces a self-motivated attitude to keep a picture of each situation constantly updated, which results in engagement. Moreover, NtoM conceives these multi-RPAS operations in civil airspace which, contrary to what may seem, could be a reality not too far in time. A SESAR outlook about the drone market [5] estimates that within 5-10 years it will be possible the adoption of operations where one remote pilot controls several aircraft. The main difference with respect to the military environment appears on the priorities: in civil airspace, where safety is a key pillar, it will prevail over effectiveness or productivity. Accordingly, in NtoM different measures will try to mitigate the possible impact that the concurrent piloting could cause on the work of the ATC or the nearby traffic. These could even provide benefits or added safety measures to a one-pilot-one-aircraft ratio. Besides the previous benefits, the framework provides the airline with a tool to maximize the efficiency of its human resources, manage the service continuity, the skill retention and the rest rules of the pilots; all this while trying to avoid the RPAS pilot burnout [21]. As the concept would be applied in the new architecture of European ATM, it should stick to its principles, some of which are to improve efficiency, capacity and security. This is how the characteristics of NtoM contribute to those aspects:

- *Efficiency*
  The framework should allow pilots to comfortably operate several aircraft at the same time. This is feasible because not all flight phases are equally demanding and many tasks are automated. If a viable load of simultaneous work per pilot at all times can be guaranteed, then the productivity of the staff multiplies. This implies an evident cost reduction for the airline due to the decrease in the number of pilots needed to exploit the fleet. In this respect, a question that may arise is if this economic motivation is well-founded and these savings on staff justify the efforts. The answer is that it depends greatly on the country. In his presentation for the Single Pilot Operations (SPOs) Technical Interchange Meeting [22], Dr. R. John Hansman, from the Massachusetts Institute of Technology (MIT) International Center for Air Transportation, presented data from 2010 showing that staff costs for US air carrier operations were approximately the 25% of total costs. Considering a 20-year service life for the aircraft, the aggregate flight crew cost per cockpit seat meant half of the cost of the aircraft. However, the percentage of the staff costs for China in 2000 —when for the US was also around 25%—, was about the 3%. During the second quarter of 2015, US airlines staff costs were the 30.65% of the total costs [23].

- *Capacity*
  Reducing the total number of pilots required should not necessarily imply a workforce reduction. It could result in an increase of the fleet and therefore of the service provided keeping the same staff size. We must also consider that the RPASs reduce greatly the bottleneck generated by the rest rules of the onboard pilots and the associated cost of the take-off and landing. By

allowing cheaper long endurance flights, the RPASs increase the availability of many applications and services.

- *Safety*
  In NtoM, the system has the possibility to constantly monitor the aircraft status and the pilot's operations. Alerts or automated emergency procedures could be triggered in case some abnormal situation is detected. These situations include pilot incapacitation, GCS malfunction or breach of orders.

- *Information*
  The possibility to provide detailed information about the status of the aircraft via data link benefits both the safety and the efficiency of the ATM operations. Automatically inform the controllers about the intent of the aircraft is of special interest when an off-nominal situation requires executing an associated loaded procedure, or just in those cases where a delay in the execution of orders or in the communications could be expected. Also, as will be explained later, the system would be able to build a map of workload from the RPAS pilot point of view, which could be of interest to analyse the integration of the unmanned aircraft in airspace.

1.3 HYPOTHESIS

The solution suggested to achieve all the aforementioned benefits expected is a network with clients and a server implementing a set of tools, policies and procedures. The hypothesis is that these elements, which address aspects like the workload, the usability or the monitoring of the operations, will allow pilots to control several flights comfortably and safely, with the proper awareness and contingency measures in an scenario like that of the civil operations. All this minimising the impact on the work of the controllers.

1.4 METHOD

The development of the ConOps followed the flow shown in Fig. 1. A concept like this, focused on human factors, must start with a deep analysis of the operations, tasks and the issues that could generate the new paradigm that should be addressed. Particular emphasis was put on the CPDLC, as this technology is the basis of part of the workload management and safety measures.

Next step was to find out the lessons learnt from the RPAS operations, which have been developed mainly in the military area. That experience has been translated into illuminating hints to address the usability and procedures for the civil application. During this phase, some guidelines were provided by a subject matter expert on RPAS human factors [20] and a military RPA pilot. Then, there was a review of previous approaches for multi-RPAS piloting, the proposals for enhancement of the CPDLC, the improvement of the awareness and reduction of the head down.

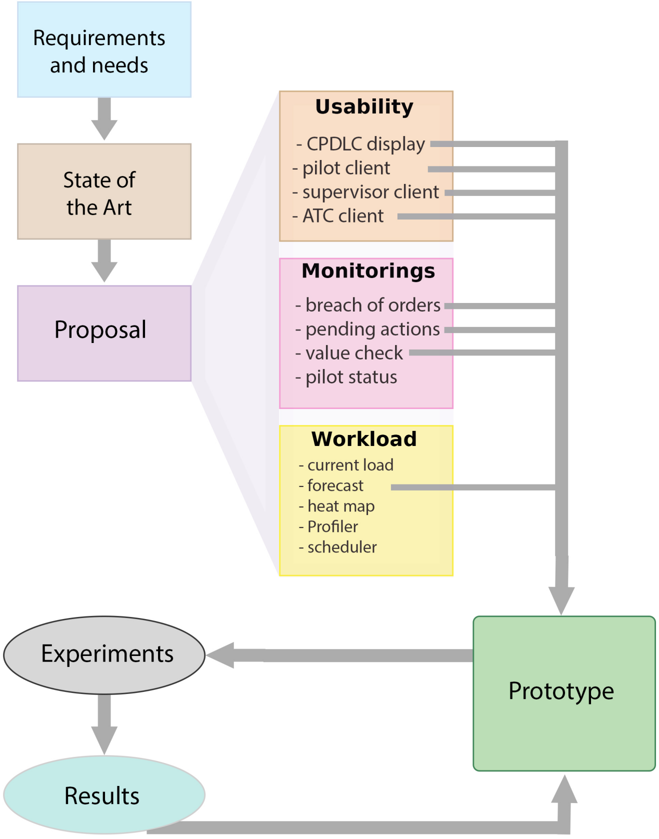


Figure 1: Flow of development of the prototype of the ConOps.

With the conclusions and background from all the previous literature, it was time to develop the NtoM framework, defining its main features and the system orchestrating it. To illustrate the potential of the concept, put to test the design decisions and discover weaknesses or improvements, a prototype of the system and the clients were implemented with part of the features proposed. It was then adapted to serve as a synthetic task environment to run experiments. These were focused on observations that could be translated to a more realistic scenario, like those related to usability, workload management strategies and subjective perceptions.

1.5 STRUCTURE OF THE DOCUMENT

Next section details the concept and features envisioned to support the objectives. Some of them were implemented in a prototype, which is described in the following section. Later, the reader can find the experiments performed with that prototype and the results obtained. Finally, there are the conclusions and a list of the remaining work that would be necessary to advance in the development of the framework.

The state of the art and the necessary background on the topics mentioned has been distributed throughout the document, placing and discussing each reference in the section of the related issue. It was considered a better way to show the comparison between the existing approach and that of the current proposal.

Some passages and figures of this document have been taken from the open access articles published to disseminate the work described here.

2 THE NTOM CONOPS

The multi-RPAS piloting is not a new concept and can be found in Air Traffic Control (ATC) training simulators or in the military environment. In fact, the supervision or control of multiple RPAS is a growing topic of research still in its early days, and several approaches can already be found. Some of them are more appropriate for specific kinds of missions, others require almost full autonomy of the aircraft, and most of them were conceived for segregated airspace. We find different paradigms; those where the tasks of the pilots are more similar to those of the manned aircraft, or ones where their task is mainly defining and guiding the mission [17]. For instance, in [16] each aircraft is not permanently assigned to a team or pilot. Instead, the fleet is considered to be able to mainly proceed autonomously. When an aircraft requires some interaction from the operator, its petition is queued until it can be served, and the aircraft waits for its turn, performing some holding pattern. While this kind of concept avoids any task overlap among flights — the operator(s) attends only one at a time, a.k.a. *vehicle-based control* — and therefore the workload is always limited, the fact that the aircraft could suddenly enter in a holding pattern to wait for the pilot's attention, something which may be unexpected by controllers, seems more acceptable in segregated airspace.

To increase the efficiency and efficacy of the pilots, much research, mainly addressed to military operations, focuses on improving the design of the GCS. Some of these improvements are based on a careful balance of cognitive resources and mental workload. Others try to enhance the human scope by teaming pilot and adaptive automation [24]. The search for productivity has also sometimes been based on task scheduling [15] [25]. NtoM combines a bit of each of those approaches: it tries to avoid unmanageable levels of workload by scheduling and task support, but also provides measures to back up the pilot if it is unavoidable; at the same time, efforts in usability and awareness try to reduce the total cognitive load of the pilot.

This section provides a high level description of the features of the framework, which are described with more detail in the section about the prototype. The section begins with the assumptions and scenarios targeted and then describes the features envisioned for the concept suggested. While the features appear included in categories depending on what are they addressed to, some measures from one category can have clear influence on others; usability affects workload as much as the support of the monitoring benefits the usability and the workload. Then, such categorisation is not exclusive and the success of a feature can have a dependence of other factors; for better or worse, a change in one of them could affect the efficacy of another. The features under these groups try to support the hypothesis that the safety of these multi-RPAS operations would be guaranteed if:

- the system is able to distribute the workload avoiding risky amounts of workload at every moment;
- the implications of the alternation of flights are addressed to avoid any situation of risk, providing the appropriate awareness;
- the concurrent piloting is supported to minimise the impact in the work of the controllers or the traffic nearby.

2.1 ASSUMPTIONS

Following is a list of assumptions regarding the terms used in the document and the conditions, scenarios and kind of operations that would better fit in or leverage the concept. The overcoming of current unsolved issues regarding the acceptance of the integration of RPAS by the general public, or the legal requirements of the States, like policies or the definition of responsibilities, is considered a mere question of time, as everything suggests ([5] estimated in 2016 that the full integration of drones in all classes of airspace would be a reality in 5-15 years). In the worst case, the concept could still be useful in segregated airspace. This work aims to be a brick in the building of the RPAS integration but considering that the foundations of this integration are just a matter of time.

Kind of Missions Targeted

One of the main potentials of the system relies in the management of the workload, which is only possible with flight plans of predefined trajectories and scheduled tasks allowing a forecast of it. For example, those surveillance missions or aerial firefighting operations in which the path and tasks are decided on the go would not leverage the concept. While the system could handle the consequences of a change of trajectory, the recalculation of the workload forecast could force a rescheduling and negatively affect other pilots, especially if done often. Then, the target missions, rather than being determined by the altitude, the kind of airspace or the capabilities of the aircraft, are those allowing a minimum prediction of the tasks of the pilot.

It must also be considered that the concept tries to leverage the periods of low workload of the pilots, which are more prone to appear in long enough flights; so long endurance flights are especially suitable to leverage the framework. This kind of flights is usually found with fixed-wing medium and large drones; for instance, in surveillance or scientific missions. Leveraging the concept applying it to fleets of small drones covering short distances for delivering tasks, would depend on the kind of mission and level of autonomy of the system. The framework has not much sense with fully autonomous drones and the fully manual operations — 1-pilot-1-flight ratio —, could benefit only from part of the features. But it could probably be useful when there is a partial autonomy or the mode changes to autonomous to manual along the mission.

Kind of Vehicles

NtoM was conceived keeping in mind the RPAS case, but the whole concept could be applied to other vehicles. Or generalising, to parallelise processes with tasks requiring constraints or preferences; processes which should be monitored and performed while keeping an adequate workload of the staff involved. Whatever the vehicle, to exploit the full potential of the concept, the level of automation or autonomy should allow the pilot to delegate a large part of the tasks, which is key to be able to control several systems concurrently. An example of this appears in [26], where a military pilot is expected to control several unmanned aircraft while piloting a manned aircraft. Otherwise, we would have a one-to-one operator-vehicle ratio, which anyway could still leverage some of the features of the system.

In the present appliance, the term multi-RPAS was chosen instead of multi unmanned aerial vehicles (multi-UAV) because the latter is usually associated to swarms of drones with a total or very high level of autonomy, cooperating to accomplish a low-level flight (LLF) mission at the same time, in the same area. But NtoM aims to help pilots control heterogeneous aircraft models, performing different kinds of missions, not necessarily in the same airspace, and having asymmetrical needs of ground crew coordination. Note that the use of the term RPAS means that the concept considers that ultimately, even with a high level of automation, there is always a human pilot in command.

Required Ground Control Stations

The framework does not include its own GCS but, if required, that used should be able to manage a heterogeneous fleet with different levels of autonomy, capabilities and equipment, or even different kinds of vehicles (terrestrial, maritime, aerial). [27] addresses the requirements and design for this kind of control station, a sample of which was part of the ambitious DARIUS project, with its Generic Ground Station able to control maritime, aerial or ground unmanned systems used in search and rescue operations [28]. DARIUS focused on the coordination of workgroups participating in emergency actions, making available to them the information collected by the different platforms. The NtoM concept could be applied also to any type of unmanned vehicle or process supervision, because the basis of the system management relies on workload, task interaction constraints and procedure definitions, which are common concepts underlying the different activities performed. However, the NtoM operator interface should be adapted to show the information and commands required to supervise each kind of vehicle; in the current prototype just RPA were considered.

Whatever the vehicle or platform, NtoM intends to be agnostic to the control station; we could have any GCS plugged into the system via the NtoM pilot interface (NPI), provided that it offered an application programming interface (API) for the NPI to be able to bidirectionally communicate with the GCS and track the activity of the operator, send commands to the vehicle or get its telemetry. And this should be possible for several aircraft at once. The availability of such GCS will be assumed.

Exploitation of the Controller-Pilot Data Link Communications

Considering that a proposal like NtoM would not have a immediate implantation, it assumes a future widespread, fully and well-established implementation and use of the Controller-Pilot Data Link Communications (CPDLC) application [29, Chapter 2.2.5], which is expected to substitute voice as the main way of communication between the aircrew and the ATC. This implies using text messages for the communications instead of voice, with many benefits for all parties [29, Chapter 2.1.1][30], like the increase of the sector capacity, decongestion of voice frequencies, avoidance of input errors by the pilot (some ATC instructions can be directly loaded into the Flight Management System (FMS)[29, Chapter 5.3.5]), the impossibility for a pilot to accept by mistake a clearance addressed to another pilot, or avoiding difficulties and errors related to translation, pronunciation and diction. No stepped-on transmissions and no need for readback. It also reduces the workload of pilots and controllers, as it simplifies the procedures and handicaps associated with voice communications a great deal. For instance, a sampling of the voice activity of the Kansas City Air Route Traffic Center (ZKC) [31] showed that 61% of the controller communications had to do with the voice frequency change procedure (32% issuing frequency changes, 29% for pilot check-ins). Using CPDLC, while the transfer of data authority (the Air Traffic Service Unit (ATSU) currently communicating via data link) is seamless to the pilot, the voice frequency change still requires some actions. In the simplest implementation of this procedure, the flight crew receives an instruction, with or without a condition, to monitor the new voice frequency (e.g. UM121 AT [position ground air] MONITOR [unit name] [frequency]), the crew acknowledges the message, and they just need to change it when indicated, with no check-in required (although the transferring controller could require it in the message). This procedure implies a lower risk of mishearing and generating read-back and hear-back errors; therefore, fewer NORDO (No Radio) problems and less frequency occupancy time (20% of the time in the [31] analysis was spent on this process).

The North Atlantic data link system supporting the use of CPDLC is called Future Air Navigation System (FANS), with a different version depending on its use in remote (FANS 1/A, through satellite communications) or domestic airspace (FANS 1/A+, which also supports Very High Frequency Data Link Mode 2 (VDL-M2) radio communications). While the testing and use of oceanic CPDLC began in the early nineties, domestic use is, at the moment of writing this article, just offered in 62 airports in the United States [32] and only for the Departure Clearance Service (DCL), on VDL Mode 0/A, with a first small set of other services envisioned to start being offered in 2019 (NextGen's Data Comm program roadmap can be found in [33]). The FANS CPDLC version is based on the Aircraft Communications Addressing and Reporting System (ACARS) protocol.

In Europe, the CPDLC version is called Protected Mode-CPDLC, used under the continental data link infrastructure Aeronautical Telecommunications Network Baseline 1 (ATN B1), also implemented over VDL-M2. The Link2000+ programme addressed its CPDLC implementation as part of the SES concept and has been operational since 2003. This CPDLC covers a wider variety of uses but has not been homogeneously implemented yet. [34] is a report of the current level of implementation in Europe, with a map of the Air Navigation Service Providers (ANSP) offering this capability. For a worldwide map of the implementation status updated to 2017, check [35]. The European CPDLC is based on the ATN/OSI protocol. Being different protocols, the European and North Atlantic CPDLC systems are incompatible.

CPDLC is just one application of the currently underused potential of the data link technology; [36] provides an overview of the applications and services based on data link. While the NtoM concept would unfold its full potential with the exploitation of the CPDLC, voice communications are also considered, as ultimately both technologies will live together to back up each other and leverage their different virtues depending on the situation. The reason why NtoM relies on its usage is that these text messages, with their predefined syntax and composition rules, allow an unequivocal interpretation of their semantics. As it will be shown, this allows the system to track the instructions received or automate tasks requested on them.

2.2 AIRCREW SIZE AND ROLES

Single pilot operations are common in manned aircraft. However, in commercial aviation and large jets, a First Officer (FO) is added to support the captain tasks or assume them in case of incapacitation. Anyway, quite research has been done [37] to consider these operations to be done also by a single pilot. Improvements in automation and communications show that it could be feasible today for nominal operations, and some ideas have been proposed to assist the pilot during off-nominal or high workload operations, trying to mimic the role of the FO.

The high level of automation of the RPASs and the GCS's facilitates a lot SPOs. Then, the size of the RPAS aircrew is mainly determined by the kind of mission. For instance, in a reconnaissance mission, we could find an Air Vehicle Operator (AVO) controlling the aircraft, a tactical operator or mission commander, and one or two payload operators (find these roles explained first person in [38]). For simplicity, this document focuses on the case of assigning flights to AVOs (that will be referred to simply as *the pilot*) performing SPOs, but the potential of the concept would allow greater flexibility for the size and roles of the aircrew. Following are some aircrew configurations that could be supported taking into account that the assignment of multiple flights could be applied to any of the roles:

- *Temporary roles*
  Previous concepts have addressed the temporary roles from different perspectives and one of the situations favouring this paradigm is the seek for implementation of SPOs. The economic interest — as shown by Boeing [39][40] — that can impel the development of SPOs seems not less important than the need to cover a growing demand for flights in the face of a shortage of pilots, a situation with an expectation of worsening in the future. In an article about the acceptance of unmanned passenger planes, we can read the following words from Alasdair Whyte: *"The world will need up to 200,000 new pilots in the next decade, and up to 790,000 over the next 20 years to*

meet this growing demand and replace those pilots who have reached the mandatory retirement age" [41].

A proposal for SPO with temporary roles was suggested at NASA[37]. On it, we find a crew of ground operators (GOs) acting as dispatchers, suggesting beneficial route changes for maybe around 12 flights each of them. But when one of the aircraft of a dispatcher needs dedicated assistance, a supervisor — a role with points in common with the NtoM supervisor, described later — transfers the rest of flights to other dispatchers and the dispatcher acts as its FO during the time required. The GO is not tied as a FO to any specific aircraft, and could play this role with any of them. An added benefit is that while performing this role, the GO has more information and tools available than from the cockpit, being able to better assist the captain (nearby traffic and weather movement prediction tools or the Emergency Landing Planner, which, in need of an emergency landing, suggests the best option between the available airports given the weather and the aircraft status). Authors explain the concept and show these tools and the Cockpit Situation Display used by the GO in the video [42].

NtoM would allow the previous temporary role allocation with a similar underlying management as that used with the pilots. However, the handoff of flights could be managed in a more automated and assisted way. For the specific case of the FO allocation, having the system access to the information and status of all flights and the level of workload of the AVOs, it could try to predict when a captain is going to require a FO. For a task distribution like the suggested by NASA's SPOs, with a multi-dispatcher, the NtoM system could advance a gradual transfer of flights that this operator was serving, as each one or these transfers require a previous handoff briefing and the whole process, if it has to be done for many flights, could last more time than the captain requiring assistance could wait for. The system could also try to balance the flights of the multi-dispatchers, to try to reduce the number of them to be transferred when they need to move to the FO role.

Anyway, same as the following configuration, the role of the AVO will not be considered for this kind of rotation.

- *Temporary assignments*
  Another dynamic allocation concept specific for the RPAS case is the one proposed in [16]. On it, the requirement for assistance comes from the aircraft itself, that proceeds autonomously until it requires the pilot to make a decision. If the pilot is busy, the aircraft executes a holding pattern until it can be attended. While this paradigm could be of interest in some scenarios and missions, it should be assessed the impact on the Air Traffic Management (ATM) in non-segregated airspace if an aircraft suddenly begins to perform an unexpected holding pattern. Another issue in this concept, which conceives the piloting as a sparse set of tasks, is the loss of awareness. Making a picture of the situation of a flight is a slow process with a high cognitive load. If the requirements of attention are very separated in time, the cost would be similar to that of a handover, something that NtoM tries to avoid as much as possible. Then, the possibility of a temporary assignment by punctual need

will not be considered in NtoM for the role of the pilot. The pilot will at least monitor the same flights until released from them; that way, when a task is required, the response time is shorter, with a better awareness of the situation. Instead, other roles, like the payload operator, could be eligible for such temporal allocation, as their task usually comes into play at certain moments of the flight, and is not critical to safety. For instance, a payload operator for several aircraft, each of them controlled by a different AVOs.

- *Multiple temporary assignments*
  An ambitious complex situation would be that where different roles of the aircrew, except for the pilot, are added and released from the flights as required. NtoM could allow such configuration considering the coordination like any other task, therefore requiring the definition of the constraints and dependencies among the flight crew roles. For instance, if a payload operator requires the involvement of the AVO to perform a task, this should be considered a task also for the AVO, with its specific requirements of time and attention.

THE SUPERVISOR NtoM includes a role to support the framework with human assistance. The tasks of the supervisor would be determined by modes or levels of automation; that way, a supervisor who is busy attending some issue could delegate the decisions to the system, which would apply predefined behaviours and policies, avoiding confirmation requests.

It could be configured which things require confirmation, individually or in groups, what will be called, for short, automation modes. A highly automated mode would be useful if the supervisor is not available or is busy attending some urgency. A mode requiring confirmation to apply each policy provides the supervisor with the flexibility to overwrite default policies in case of need.

One of the tasks of the supervisor will be to provide an optional human confirmation or selection of the suggestions of the system when it tries to manage situations. Supervisors could force assignments and transfers, overwriting the scheduler decisions, or create stages and constraints. For a large fleet, the supervisor could be assumed as the operations manager of the airline operations centre. In smaller fleets, this role could be mixed with the one of the dispatcher, assuming other tasks like requests for feedback (the supervisor has a chatroom available with every pilot), clarifying actions and assisting pilots. Besides the usual information that an operations manager has available, the NtoM supervisor interface would provide: the dynamic representation of workload forecast ([43] screencast 1), the history of actions and communications, the sequence of pilots for a flight and the log of any relevant event like warnings, lost links or emergency procedures loaded.

Therefore, the supervisor interface will be oriented to manage the assignments and monitor the feedback and activity of the pilots. Its flight representations should depict the critical, urgent or pending situations and the supervisor should be able to quickly change to a full or partially autonomous mode if is in need to attend some issue. Scheduling suggestions should show different alternatives

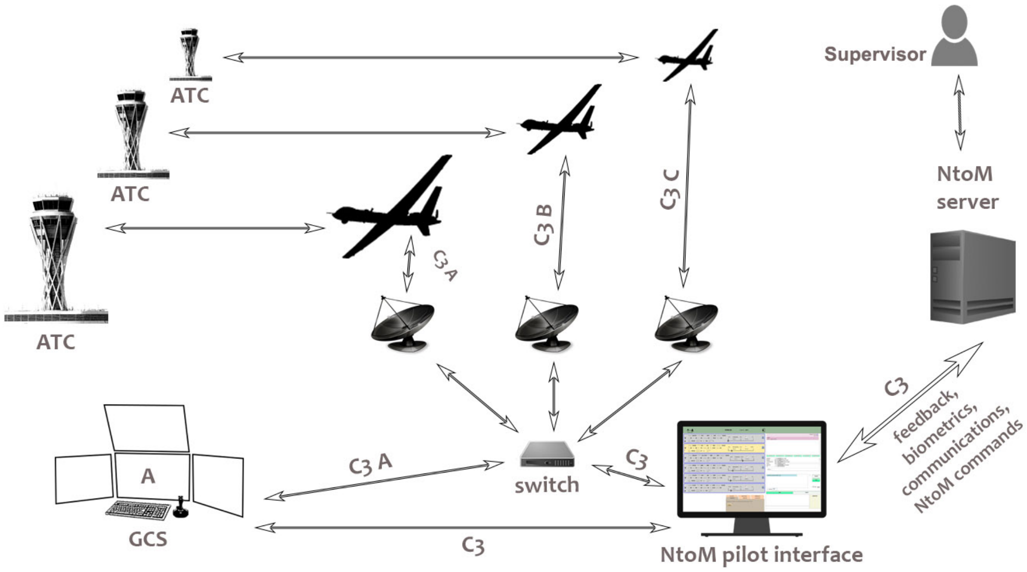


Figure 2: Connectivity schema of the aircraft and the GCS to the NtoM network.

ordered by the number of preferences that the pilot meets and their current workload, to allow an informed decision.

2.3 SYSTEM ARCHITECTURE

Fig. 2 shows how the different elements connect to the NtoM framework. The GCS is able to support multiple Command, Control and Communications (C3) links, as either an embedded capability in the GCS, or provided by an external switch. An example of a GCS with this capability is General Atomics' Heresy multi-mission control, a software that a single pilot can use to control up to six large UASs like the MQ-9 from a common laptop including the possibility to transfer the control to other pilots in the network [44]. Depending on if the GCS has this multi-link capability, the access of the NPI to the C3 would be via the GCS API, or directly connecting to the switch.

The GCS is considered as an independent piece plugged into the system via the NPI. In those GCS's where, while being able to keep the link with several aircraft, only one is controlled and seen at a given time, pilots would select one of the flights in the NPI to load it (concretely, give the focus to it) in the GCS. The role of the NPI is to facilitate the task of controlling concurrent flights and detecting possible errors. This role separation is similar to that found between the Playbook-enhanced Variable Autonomy Control System (PVACS) and the Geneva Aerospace's Variable Autonomy Control System (VACS) [45]. Like PVACS, the NPI is a high-level tool for multi-UAV management; and VCAS GCS could be compared to the GCS plugged into the NPI, providing a library to execute control commands. In both cases, the integration becomes a symbiosis: the higher-level part needs the lower-level interface to send control commands for the aircraft and receive status information, but the lower-level part benefits from the automation and planning that the higher-level part provides.

2.4 INPUTS

FLIGHT DEFINITIONS Dataset containing information required for the system to manage and monitor a specific flight, like the flight plan, scheduled tasks, aircraft equipment and capabilities, ground crew coordination, payload, emergency procedures or preferences and constraints to be considered for the candidate operators of this flight in particular.

CONSTRAINTS AND PREFERENCES Constraints are adequate, for instance, to reflect authorities regulations, like the type of license or certificate that the pilot should hold to be in charge of an aircraft with certain capabilities or a kind of mission. The airline could also define its own constraints, like working and rest hour distribution. Preferences would be those assignment criteria assumed as soft constraints, giving points lead to those candidate pilots remaining once applied the hard constraints.

STAGES Apart from the preferences concerning the whole flight, other ones could apply during portions of it. Those portions have been called stages (not to be confused with flight stages). The term wants to reflect that is a piece of a whole of indeterminate elements, with some sense of sequence or order, while not related to the criterion followed for the fragmentation. These allow to prioritise pilots specialized on the tasks contained (functional migration [18]); or just the opposite, they can move up those pilots with less experience to help them train or keep skills. The definition of the criterion for being eligible would be of any kind that the scheduler was prepared to handle: a nominative list of pilots, total flight hours, a measure of experience with the involved tasks, etc. The criterion to delimit a stage could also be based on flexible references. Some examples are:

- *Stages limited by waypoints*
  Containing all the legs of the flight plan between two specified waypoints (Fig. 3(a)).

- *Stages representing periods of time*
  Priority to choose certain pilots in a time slot of the flight (Fig. 3 (b)), or references like "in 20 minutes after the waypoint x", "one hour after taking off", "during 3 hours from waypoint x", etc.

- *Fuzzy stages*
  The beginning and/or end of these stages is subjective or is not predictable or measurable, therefore it is decided by the pilot or the supervisor during the flight. For instance, they could delimit flight phases (Fig. 3 (c, upper)), the area overflown (Fig. 3 (c, lower)), or even the meteorological conditions. These stages would not be taken into account during the pre-flight schedul-

ing, only during the online planning, once the pilot or the supervisor sets that the stage is active.

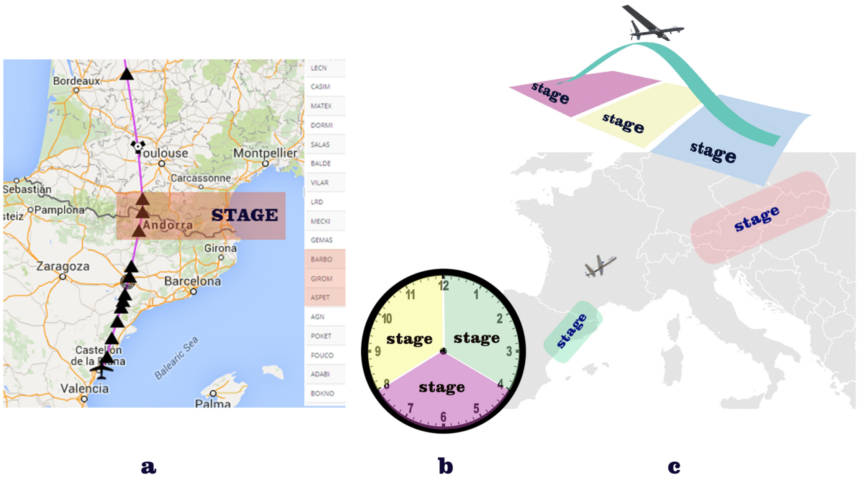


Figure 3: Types of preference stage delimitation: (a) by waypoints; (b) by time; (c) limits established during the flight.

Stages can help the airline to more favourably organise the staff timetable. These could determine the pre-flight scheduling but also be defined dynamically during the flight. Different policies to follow when entering a stage during the flight could be applied by the system. An option would be, when reaching the stage, to ask the scheduler to find an available pilot meeting the preferences if the current one does not. Another one would be not forcing a handover, but the scheduler would take into account the preferences if a migration of control is necessary during the stage. The choice would depend on the criterion of the airline to give more importance to fitting to preferences or avoiding handovers.

2.5 MANAGEMENT OF THE WORKLOAD

Caring about the workload of the pilot is an obvious safety measure; an overloaded pilot could not be able to operate or monitor all the flights with no risk. Also, there are regulations that address the maximum workload on individual crew members to determine the airworthiness of the aircraft (e.g., [46] FAR Title 14, Part 25.1523 Minimum flight crew: *"The minimum flight crew must be established so that it is sufficient for safe operation, considering (a) The workload on individual crewmembers; (b) The accessibility and ease of operation of necessary controls by the appropriate crewmember [...]"*). While this example is addressed to the maximum workload when piloting a single aircraft, the NtoM system should guarantee that the same applies to the case of being piloting several aircraft concurrently, that the total workload is kept under a safety threshold at every moment. Besides, the assignment of flights should try to be engaging — not too low — but not too high because, maintained in time, a high and unbalanced load could contribute to occupational burnout.

Controlling several flights is possible because many current RPAs are highly automated and not all flight phases are equally demanding, sometimes reduced to long and monotonous supervision [47], which leads to disengagement and loss of awareness. In this respect, the multi-piloting induces a self-motivating attitude of awareness (Yerkes-Dodson law), as could be checked during the first set of experiments when the performance of the participants and the response time to ATC were better when they had four flights compared to having just one.

But the system must guarantee a viable load of tasks per pilot at all times. To that end, it needs to predict, evaluate, limit and distribute the workload. This responsibility makes the workload management a key aspect of NtoM. This section explains how it will capture, represent, manage and predict the workload of the pilots.

2.5.1 Scheduling Before the Flight

When the assignment of the flights is decided before the takeoff the system will play the role of a workforce scheduler, helping the airline operator to manage and optimise the staff timetable. It will suggest the best assignments of flights considering constraints and preferences, but also a viable workload at every moment considering the development of the mission as projected in the flight plan. This is not just a fancy feature, but could also mean substantial benefits for the company. As Dr. R. Mike Norman, of the Boeing Company, pointed during NASA's SPO Technical Interchange Meeting [37], when considering the crew costs relative to the number of pilots, the company considers about seven pilots per seat — captain or FO —, due to pilots' rest rules, training, etc. He suggested more efficient scheduling to reduce the number of pilots per seat considered and save costs.

The pre-flight scheduling handles a timeline for each flight, representing the forecast of workload, with the planned tasks placed along at the moment they are expected to be executed. For a same pilot, if the tasks of different flights coincide in time, the fact of being a forbidden task overlap or surpassing the workload threshold of that pilot will be hard constraints to not allow the assignment of involved flights. The Scheduler of the system, given a set of flights and pilots, will try to find a combination of assignments of flights to pilots in a way to minimise human resources while keeping a safe level of workload at every moment. The forecast is built not only placing the peaks representing the workload of planned tasks, but also with the available information of events that could increase this workload, which will be detailed later.

2.5.2 Scheduling During the Flight

Several situations could require a control migration during the flight. The transfer could have been scheduled due to the end of the workday of the current pilot, the beginning of a preference stage or because of coverage issues. Pilots can also ask for a release of specific flight(s) if they consider that their current workload is excessive or they are having any contingency that requires focusing on the involved aircraft. They are provided with this feature because the prediction of the cognitive workload is not an easy task, and is at the mercy of the emotional or psychological situation; allowing pilots to transfer flights acknowledges the limitations of the system and places on the subjective experience of the pilot the importance that it requires. Anticipating that, a transfer could also be a decision of the system, as it would be able to keep track and predict the flow of workload and could force a reallocation of flights if it considers that the pilot should be released of load or there is a conflictive overlap coming. The supervisor could also force a transfer. Another case of migration could be the result of a contingency measure; for instance, if the ATC instructions are disobeyed or forgotten, and the pilot does not correct or justify his/her actions after being requested. Finally, if there is no activity detected from the pilot for a while and the requests for feedback do not get an answer, an automated rescheduling and transfer would be forced, as it could be assumed pilot's incapacitation or jamming.

When any of the mentioned situations occur, the scheduler of the system will act as an online planner, trying to find the most appropriate pilot to receive the flight, similarly as in the pre-flight scheduling, but this time only considering those pilots currently logged in the system. First step will be to discard those pilots who do not meet the constraints to pilot the flight. Then, the workload forecast of the transferred flight is overlayed to those of the flights each pilot is already controlling, to identify any overlap incompatibility of tasks or excess of workload at some point. The assignments passing this filter, would be ordered from best to worst, based on the preferences that they meet and the balance of total workload among pilots.

The best candidate could be automatically assigned or, depending on the automation mode, the ranked list showed to the supervisor for selection. A similar dynamic crew allocation and supervisor role can be found in [24] when the dispatcher needs to hand off any flight to offer dedicated support as a first officer and the supervisor is in charge of the re-assignment of fligths; in NtoM, the best candidates would be automatically calculated and the supervisor, if present, just needs to select and apply a suggestion.

2.5.3 Sources for Workload Estimation

To evaluate the current and the forecast of workload at every point of the flight, the system would make use of as many sources of information as it is provided with. Following are suggested some of them. Their total contribution should be determined guided by subject matter experts or analysing their repercussions on the performance:

- *Flight definitions*
  The first clues to place tasks and their associated peaks of workload are provided by the flight plan. The moment to execute the scheduled tasks could be associated with references like waypoints, time, altitude or a mix of them. During the pre-flight scheduling tasks are placed in the timeline based on the intended performance, like planned speed, therefore providing a first rough forecast. But when the forecast is updated during the flight,

these peaks are placed based on the current telemetry of the aircraft, which provides a more plausible estimation.

- *Communications*
  During the flight, ATC instructions can add tasks, make scheduled tasks vary or just disappear. An example of this can be seen in [43], Workload section, Screencast 2. Not all communications from the ATC require an answer from the pilot; some are just informative or only require an acknowledgement. Then there are requests for reports, negotiations, or instructions that should be followed immediately or when some condition is met. CPDLC allows to easily discriminate each kind of communication and instruction, and therefore allocate the workload that each of them implies.

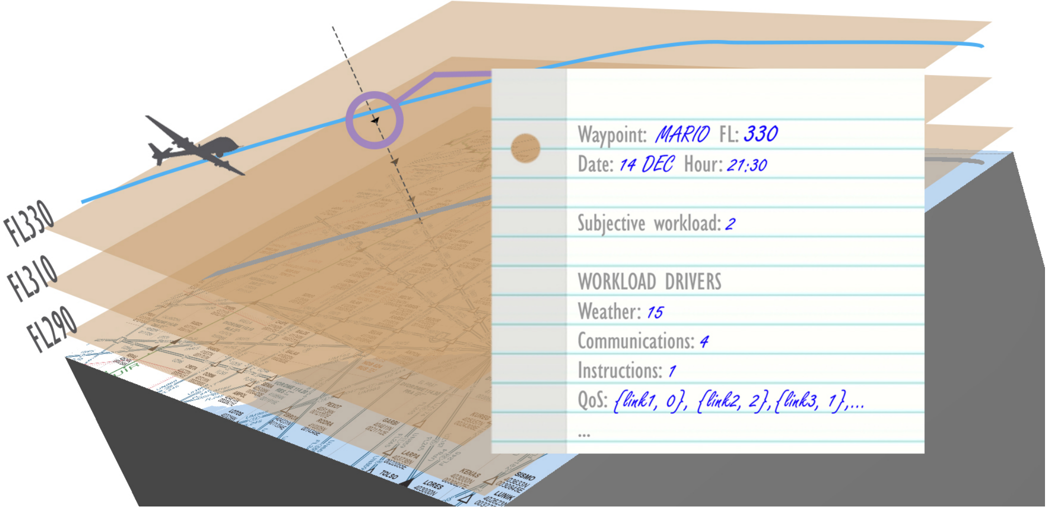
- *GCS activity*
  The monitoring of the pilot activity allows the system to detect when a task has been executed or if it has been delayed. This allows to delete or relocate the peak of workload associated.

- *Aircraft Status*
  During the flight, aircraft readings can have an impact in the evaluation of the workload. For instance, mechanical and electrical failures in the aircraft could mean an increase of workload, maybe requiring exclusive dedication of the pilot to the flight.

- *Link quality*
  Same as the status of the aircraft, an unexpected degradation of the link's QoS could require to make some tactical decision, increasing the workload.

- *ATM Reports*
  Meteorological conditions and forecasts, traffic density, NOTAMs (Notice to airmen) or any service of information that could affect the development of the flight, like the envisioned System Wide Information Management (SWIM) [48]. Some of that information is also available while planning the flight [49].

- *The Workload Heatmap*
  One of the suggestions of the NtoM concept is the elaboration of a multidimensional map reflecting the expected workload from the specific RPA pilot perspective. It would be consulted by the Scheduler to check the possibility of an increase of workload along the trajectory. The idea is that the workload drivers associated with a waypoint should be acceptably stable and predictable if we consider the extra variables of altitude, time and date. It would be initialized with historical data, like those we can find in [50]. These data should represent any kind of event that could give a hint about a possible added workload for the pilot, like ATC interaction ([51] uses machine learning to predict the level of workload from the controller-pilot voice communications), weather, traffic density, link quality, subjective assessments or biometrics. In short, not task-related workload, but circumstantial workload or the pilot's perception of it.



Waypoint: MARIO FL: 330
Date: 14 DEC Hour: 21:30

Subjective workload: 2

WORKLOAD DRIVERS
Weather: 15
Communications: 4
Instructions: 1
QoS: {link1, 0}, {link2, 2},{link3, 1},...
...

Figure 4: The workload heatmap would register the expected contributions of the workload drivers and the subjective experience of the RPAS pilot for each coordinate *{waypoint, flight level, day of the year, hour}*.

The database describing the heatmap would then be composed of a set of tuples *(waypoint, altitude, hour, day_of_the_year, averaged_workload)*. While the granularity of these values should be surveyed to pilots and subject matter experts, the meaning behind them is as follows:

- *Waypoint*. The geographic point to which the data makes reference. How many of them to represent an area and which ones are questions that should be consulted.
- *Altitude*. The flight level also determines the traffic density and therefore the possibility to receive instructions from the ATC. To this respect, one of the decisions before the sampling should be to decide the altitude ranges that each tuple represents and if those ranges should vary depending on the waypoint.
- *Hour and Day of the Year*. The periodicity of the temporal sampling could be decided by searching and selecting sufficiently different patterns of workload in the historic data. Seems clear that the workload imposed by the traffic will vary along the day. But it should be checked if 24 samples per day are enough or it should store a tuple, for instance, for every half hour. The traffic of a Wednesday will probably be different than the one of a Friday. And we can expect a big difference between a week in February and a week of the annual summer exodus, that is why the top periodicity is suggested at a day-of-the-year level.
- *Average Workload*. Represents the average workload expected in that point (in practice, a surrounding area). This data would be a set of probabilities of the different workload drivers to appear. That way, it can be tuned the final impact in the total workload of each driver; for instance, traffic density, even with a high probability of being high, could

have less impact in the total workload than bad weather or certain ATC instructions.

- *Perceived Workload*. This value, when available, would collect the real experience of workload from the pilot point of view.

While this map could be initialized with historic data associated to manned aircraft, it would be updated with those values registered by the RPA pilots while overflying the location of the sample. Those new values would be based on the track of the events, communications, performance or in the biometrics provided by any wearable device. The aggregation of the experiences of these pilots would draw their perspective about workload in the area, which could differ from that of the manned aircraft due to the level of automation or the specific procedures or even airways that could be assigned to the RPA in a future full integration. The reason to not building the map from scratch, with default values, is that this could cause high unexpected peaks of total workload due to the inconsistency between the default value and a high real one. Then, is preferable to initialize the map, even with rough data, in a way to reduce the possibility to find an unexpected peak. If the manned aircraft records do not provide a close picture of the RPAS pilot experience, at least, selecting the fields wisely, these could provide some hints. Anyway, being the scenario of this concept still far in time — while probably not so far [52] —, it is difficult to guess how the integration will be carried out, and how exactly the historical data could be leveraged. The impact of a full implementation of CPDLC should also be taken into account when using those historic data, as it automates or allows to automate some procedures that currently imply communication tasks for the pilots.

2.5.4 Evaluation and Representation of the Workload

For the system to be able to balance, manage and predict the workload, it needs a representation of it, besides the thresholds and constraints to be considered. The representation will be based on the requirements of time and what will be called the *level of workload*. This level of workload will be comparable to the level of attention, cognitive load or stress measured while executing the task. This abstraction allows the use of different measurement methods, but ultimately it should represent the dedication required or the possibility to handle other tasks concurrently. The resources required by each task will then be described as a set of subtasks, each one described with a pair *(level_of_workload, time)*, where the time represents the expected number of seconds to perform the subtask. The workload level will be a single constant value along the subtask. The highest level will correspond with especially demanding or delicate executions requiring the whole attention from the pilot; here we can include the takeoff, landing and those situations where the autopilot is not an option or there is some non-nominal situation. The lowest level would be the periodic monitoring of the telemetry of the aircraft. The reason behind the simplification in a single value is to make NtoM procedural and GCS agnostic; it does not need to know exactly how the task is performed, just its duration and the possibility of parallelising more tasks.

This provides greater flexibility to the pilots. With a representation too close to the measurements, the scheduler would assume that the actions of the task are always executed at the same moment, or in the same order, which would result in a tighter straitjacketed scheduling. This would not fit with real practices, especially during task overlaps, and would work against the pilot's acceptance. The reality is that the actions of a task have some degree of freedom on the moment to be completed.

For a particular task, the resources required could vary greatly depending on the GCS but also on the subjective perception or the accumulated experience; that is why each pilot will have a personalised representation of their execution of tasks and taks overlaps, which will be called the pilot's workload profile (Fig. 5).

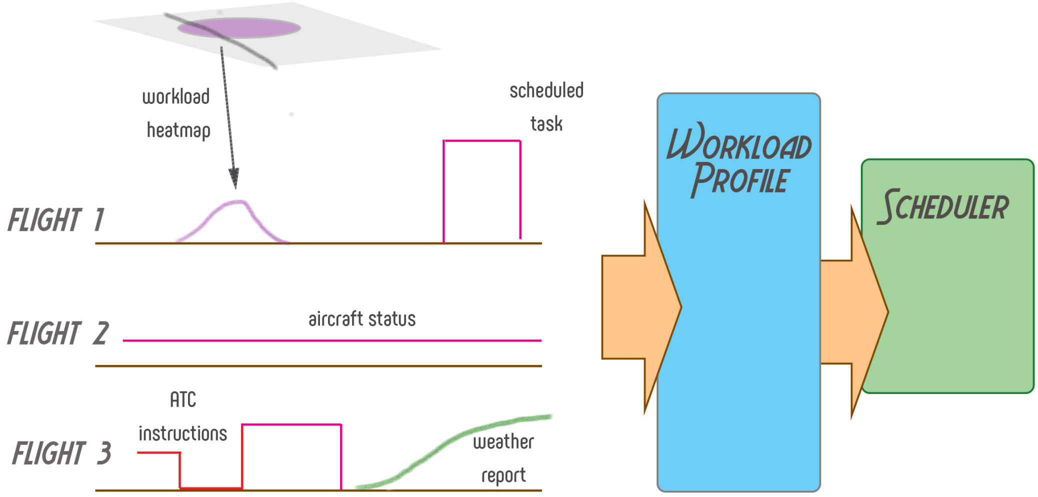


Figure 5: The workload forecast will be built with all the information available about pending tasks and workload drivers. With it as an input, the workload profile returns the estimated capabilty of the pilot to bear with the tasks. Based on its verdict, if necessary, the Scheduler could try to find an alternative assigment of flights or, if that option is not possible, ask the Event Handler for an assistance from the system to help the pilot to deal with the excessive peaks of workload.

A question that may arise is what exactly constitutes a task. Its definition is irrelevant to the NtoM scheduler, but an appropriate level of atomization should be chosen. On the one hand, not too wide to be counter-productive, not being able for leveraging the workload optimisation. In a clear example, consider a hypothetical multi-element message with a conditional clearance like:

UM20 CLIMB TO [levelA]
UM25R AFTER PASSING [position] DESCEND TO [levelB]

If the pilot replies with a WILCO and the whole message is considered as a single task, the time between the execution of both instructions would have a workload level associated (Fig. 6, solid colour rectangle), limiting the number of tasks that could be interleaved during that period, which is specially unproductive if that period is long. A better choice would be to consider the execution of the instruction as two separate subtasks, with their respective required resources (striped rectangles). However, it could not be the case if both tasks are too close to leverage the insertion of another, considering its duration plus the time associated to the cognitive cost of the task or flight swap. On the other hand, the atomisation of the actions that constitutes a task should not be too fine to avoid being too GCS-specific and an overfitting that could hypothesise too much about the pilot's individual workload strategy, as it could change depending on the situation.

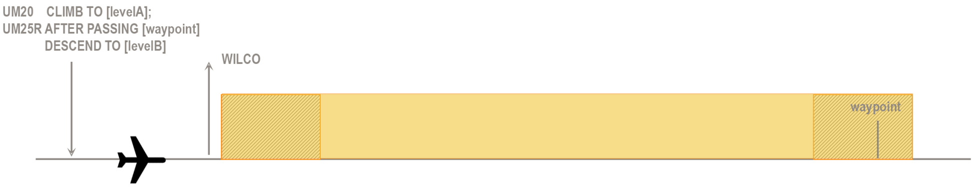


Figure 6: The determination of what constitutes a task does not need to rely in an immovable definition, seeking instead the optimisation of the productivity. Here, the message would be divided into two subtasks (shaded areas).

We can find in the literature other approaches to determine a taxonomy for the tasks of the operators. [53] shows an example of task analysis, taking the whole mission as the root of a tree that iteratively branches (e.g. mission, phase, segment, function, task) up to the sensory resources needs. [54] also exemplifies a thorough cognitive task analysis for a considerably more demanding military scenario and mission to guide the design of the GCS, finding the functional requirements and information required for an appropriate awareness. In NtoM, a task definition does not need to fit a consistent conceptual fragmentation like these, although a similar approach could help to define the tasks. The reason is that while the previous examples seek to optimise the performance in highly demanding military situations, NtoM focuses on the convenience for a safe scheduling. Then, it roughly defines a task like a set or pattern of actions requiring a rather predictable amount of time, without too long waiting times in between that could justify breaking it into subtasks to leverage those idle times. Therefore, the resources requirements of a task could change depending, for instance, on the usability of the GCS or the LOA of a particular aircraft.

The NtoM representation and management of the workload holds the qualities of sensitivity, continuous, predictive, context-rich and non-intrusive. An example for the qualities of continuous, predictive and context-rich can be found in an adaptive associate system for a cockpit that monitors the state of the pilot to predict his/her future state, and adjust the pilot's workload by providing automated support or re-allocating tasks [24] [55] [56]. In NtoM the sensitivity appears in that it provides a description of each task or task type. The context-rich quality is in the fact that the resource requirements of the tasks vary depending on whether other tasks are being executed concurrently [57] or there is information available about workload drivers via external reports. The continuity would be provided by the fact that the workload forecast is constantly updated (in the prototype every 10 s), because the expectations of workload and the position of the tasks calculated during the pre-flight scheduling will almost certainly change due to a variety of inevitable factors like speed changes, deviations from the original flight plan, weather or ATC instructions. Also, during the operations, whether relying on biometrics or analysing the performance, the system would reflect the real experience in the workload timeline; for instance, postponing the peak of a task which execution is delayed or increasing its level of workload if the biometrics reflect a higher stress than expected.

This representation of the tasks and task overlaps placed in a timeline with a timespan that varies depending on the concurrent tasks, may recall those task interference models which determine the overlap incompatibility depending on the time required for the concurrent tasks and the time available, like the Single Channel Theory [58]. But these models consider all tasks as equally demanding and would avoid overlaps of long low-demand tasks which could be perfectly executed. Weighting the tasks and the overlaps considering the cognitive load, the present approach becomes more similar to the Single Resource theory [59], which determines the difficulty of the tasks as determinative of their concurrency.

2.5.5 The Workload Profiler

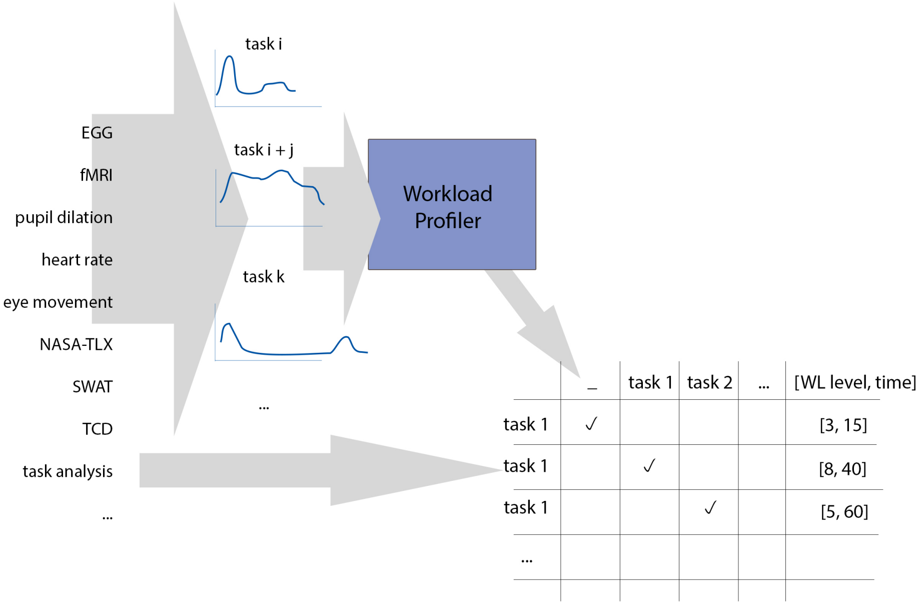
The determination of the previous workload model begins with the measurement of the activity of the pilot, which will be transformed into the representation suitable for the needs of the Scheduler described in the previous subsection. The closer the measurement to the real operations, the better prediction and scheduling achieved. These measurements will reflect the impact in the workload of what are called workload drivers in [60], where the factors affecting the workload of the remote pilots are dissected. A trustworthy representation would take into account as many environmental drivers as possible and would reflect the GCS usability, aircraft capabilities, procedures, the LOA and the pilot performance profile.

Several previous studies address the modeling of the workload of the pilots, even for the specific topic of multiple UAVs [24]. These are mainly addressed to provide clues for the system designers during the initial stages of development of the system or for the optimisation of a detected source of high workload or task parallelisation incompatibility. The workload is usually described and analysed as the allocation of resources, being these the requirements of attention, cognitive processing or the sensory and psychomotor needs. This fine-grained approach fits well with the aim of NtoM to reflect the workload behind different combinations of aircraft, GCS or procedures. Having similar resource requirements in different environments the operator performance measurement in one scenario can be assumed in the comparable case, reducing the number of measurements required. An example of a tool for this dissection of the workload is the U.S. Army Research Laboratory modeling tool Improved Performance Research Integration Tool (IMPRINT) [61], based on Wickens' multiple resource theory [62]. It helps system designers to analyse the impact on performance and time duration for each task type when tasks are executed in parallel; determines the quantity of parallel tasks that a pilot can handle at a given time and the incompatibilities among these considering the visual, auditory, cognitive or psychomotor resources required; it can also build the pilot's personalised workload profile. To evaluate the workload, during the flight simulation, IMPRINT sums the need for each resource across tasks. By contrast, NtoM sums the workload of each task across flights. IMPRINT is oriented to the design of GCS's and procedures, while NtoM builds on top of these.

One way to determine the mental workload is using analytical methods (modeling). To this respect, in [63], the task workload and the total workload of task overlappings are estimated from analytically generated workload profiles (i.e. models prepared by an expert based on the experience of the pilots). And [15] provides a mathematical model for these operator-specific requirements of time and workload to perform a task, besides a thorough review of the literature on different aspects of the topic. The model is based on task workload levels that are subjectively rated and is suggested for the dynamic assignment of tasks to aircraft and pilots, optimising the distribution of tasks on the go. This concept schedules tasks, while NtoM schedules flights. But the way the total workload is calculated at a given moment also takes into account the switching cost or cost of concurrence between tasks, something that in NtoM could be separated into two different values: the cost for the in-flight task swap and the cost for the flight swap; this differentiation comes from the especially high cost of the flight swap. Another difference is that the previous paradigm relies the scheduling of the tasks partly in the effectiveness of the pilot, while NtoM, addressed to civil operations, prioritises aspects like workload balance to avoid risky situations and the impact in the work of the ATC.

Empirical methods to determine the workload are more frequently used. These include: physiological measurements like electroencephalographic activity (EEG), functional magnetic resonance imaging (fMRI), near infra-red spectroscopy [64], positron emission tomography, eye movement, pupil dilation [65] or heart rate; behavioural measurements, which analyse the impact on performance when the level of workload varies; subjective measurements via during experiment or post-experiment surveys, like the subjective workload-assessment technique (SWAT) or a mix of these techniques. Surveys may seem the cheaper and easier method, but to avoid interfering with the work of the pilot these should be carried out after simulations, which makes the evaluation less precise. Considering that the measurements must be updated when conditions change, this means taking the pilots away from their duty every time. In this regard, biometrics provide advantages, especially those registered using devices that do not hamper the activity of the pilot, like belts, cuffs or cameras. These devices can be integrated seamlessly into the usual activity of the pilot and the tasks being measured as they appear. Another interesting benefit is that physiological measurements can reflect the impact of the accumulated time, which would allow the system to predict fatigue to reduce the cognitive load threshold.

The component responsible for building the pilot's workload profile from the measurements, simplifying them in pairs of time and level of attention required, has been called the Workload Profiler. It could receive as input the workload level of a task as a single value, for instance, when the evaluation is done by surveys — in that case the values directly populate the table (Fig. 7) —, or as a variable value throughout the task, as provided by biometrics. In any case, the measurements must provide, for each task, the time required by the operator for its execution. Once the values are set for each kind of task, the Workload Profiler will determine the allowed task overlaps and initialise their values in the table with a worst-case allocation of resources. The task overlap incompatibilities would be determined based in the capability of the pilot to assume the total workload of the concurrent



	_	task 1	task 2	...	[WL level, time]
task 1	✓				[3, 15]
task 1		✓			[8, 40]
task 1			✓		[5, 60]
...					

Figure 7: Acquisition and initialisation of the pilot's workload profile.

tasks but also applying other constraints, for instance, based in normative or just common sense (don't allow the overlap of two delicate tasks even when the sum of their workload levels do not exceed the threshold). The result would be a kind of multi-tasking conflict matrix, similar to the one defined in [24] at a GCS level for the incompatibility of resource overlapping among parallel tasks.

The aforementioned allocation of resources for a worst-case is in line with the philosophy of prioritising safety over productivity. But at the same time, it eases the pilot acceptance allowing a scheduling that respects the pilot's workload management strategy selected. This can be seen clearly in the following example. Consider a workload level measurement using some objective method that registers the level of attention during the task execution (Fig. 8, top, curves). As said, a task can contain one or more subtasks and waiting times. Except for the first one, the rest of subtasks can vary their starting moment to some extent. An example of a task could be the accomplishment of a conditional ATC clearance, being the first subtask the evaluation and reply to the message and the second subtask the execution of the instruction. In the illustrative curves of Fig. 8, each peak would represent a subtask (named *action* in the figure), with an amplitude that depends on the time required to be executed and a height of crest reflecting the stress, cognitive load, attention or whichever workload indicator was chosen. Then, the top row shows task 1 with a couple of subtasks. Suppose now that a second task arrives at another flight while the first subtask in the first flight is being executed, an event depicted by the row under the previous one. Consider that both tasks are instructions from the ATC that need to be evaluated and replied using data link (action 1) and then executed when possible (action 2). The following rows below describe different strategies that could be chosen to cope with the task concurrency. Strategy 1 portrays the human tendency to finish the present task and then follow with the new one to avoid the concern of having pending issues. In addition to this reluctance, there is the fact that pilots are aware that moving to a different flight implies recalling the situation of that flight, with its added cognitive cost (swap area). The participants in the experiments were instructed to give priority to reply to the ATC over executing an instruction that was already accepted, to avoid delaying the controllers. Strategy 1 does not follow that rule; the first task is completed, and after the time required — physically and mentally — to swap to the other flight, the pilot starts the second task. In the example, it can be seen that the moment to begin the second task almost coincides with the expiry of the margin of time to reply the controller. In CPDLC this is the time available to reply an ATC message, after which it must be discarded, which would force the ATC to swap to voice to clarify the problem, overworking the controller. Strategy 2 would be more desirable because it prioritises the reply and despite its two flight swappings it allows execution on time.

An explanation is required about what exactly means in the previous explanation to answer on time and to execute on time. In CPDLC, the air system timeout are the seconds that the pilot has to reply before the message is automatically closed (100 s as per [29]). While that timeout has an standardised value to *technically* close the message, it should not be considered a fixed value for every situation. For example, while using voice, when controllers require a quick answer (like in an area of high traffic density), if they do not get a reply in about 10 s, they repeat the communication. Considering that one of the goals is not increasing the workload of the controllers due to delays, the system should try to decide when and how to determine those margins. Similar approach would be applied to the margin for execution.

But following with the allocation of resources, the Workload Profiler would not consider Strategy 2; it would choose a more conservative, worst-case option: Strategy 3. This worst-case strategy is that requiring more time; in the example, the subtasks of both tasks are interleaved, which places a swap cost between actions. The allocation of resources for a worst-case scenario encompasses the requirements for any strategy selected by the pilot and provides some margin in case a problem or unexpected delay appears, but at the same time does not uselessly reserves excessive periods of time.

Now, looking at the Strategy 3, it seems there is no room for a third task. But it could arrive at time 17, just after the first task has been finished. In this situation, the time constraints allow the pilot to finish subtask B of task 2 and then attend to the incoming request. Therefore, the conflict matrix will have to reflect different kinds of overlapping for the same participant tasks, because not all timinigs of an overlap imply an incompatibility. As might be expected, the number of combinations could explode easily if many different tasks and combinations are considered. A way to reduce them would be by classifying the tasks by type according to their number of actions and cognitive effort, like in [66] for the manned case using the physiological measurements registered for the task. Another option to consider would be a mix of conflict matrix and mathematical model, where the formula would describe those combinations that can be calculated from the set of variables describing the overlap and the tasks involved.

Then notice that there is no intention to represent how or when the tasks are performed exactly, especially when they overlap; that is left to the pilot's will. The

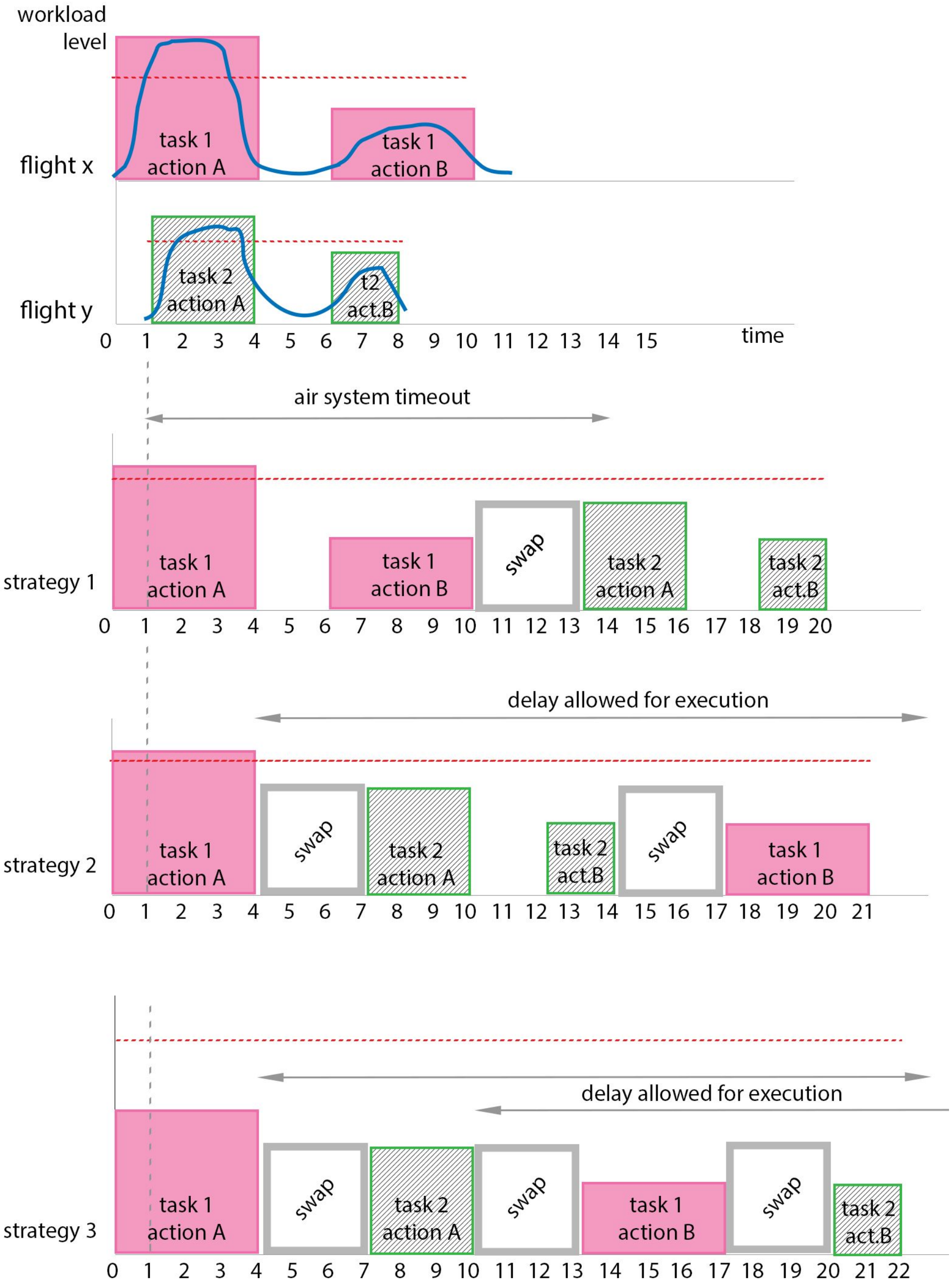


Figure 8: Strategies of resources allocation for an overlap of two tasks.

main interest is to determine the extent of availability of the pilot to handle more tasks. This respect for the pilot's personal strategy for the workload management not only follows a guideline about RPAS interface usability for pilot acceptance, it also means an abstraction that decouples the representation from any specific GCS or procedure, making it possible to easily adapt the concept to heterogeneous vehicles, mission-specific tasks, and levels of automation; the scheduling focuses on the constraints rather than in the particular executions.

The workload profile must also include the maximum workload threshold for the pilot. This limit has been called the pilot's red-line of cognitive workload [67] [68], and represents the amount of workload that a pilot can handle within an admissible decrease of the performance. It may vary for each pilot ([69] illustrates this for the ATC case), and it can increase as experience is gained and the pilot is able to manage more tasks. This threshold will act as an added constraint; a task overlap can seem feasible in the sense that all time constraints are met, but that does not mean that the pilot will be able to deal with it correctly if the total workload level is over the pilot's capability of management.

The design, implementation and testing of the Workload Profiler could include greater flexibility than the previous example. For instance, the possibility to specify precedences among tasks or subtasks. Another example could be tasks with a [level, time] value dependent on an input parameter; a use case of this feature would be to reflect the impact of the fatigue in the requirements of resources. Annex B shows a first version of the Workload Profiler.

2.5.6 Update of the Workload Profile

Using personalised workload profiles instead of the same model for all pilots provides a description of the relationship between task load and mental workload for that pilot [69]; that means the possibility to detect those skills in which each pilot stands out (also those that should be trained more) and how these evolve as the pilot gains experience. Recent experiments [64] found a decrease in the brain activity of UAV pseudo-pilots as they gained experience with the simulator, which implies a lower subjective workload. It seems clear that the full potential of a pilot would be underused if the improvement of skills, or the capacity to assume a higher workload, is not reflected in the workload profile. In the work mentioned, this improvement was detected by functional near-infrared spectroscopy.

In NtoM, the way to detect the evolution of the requirements of time and cognitive load would depend on the capabilities of the system to measure that progress. If the pilots could periodically work using a wearable (like a wireless EEG headset, a chest strap or a wrist-worn heart rate monitor) to record their physiological response or other non-invasive methods like cameras to track pupil dilation or thermal imaging [70], then the process would be quite similar to the initialization of the workload profile. If the system does not have access to these measures, it could try to infer the skill improvement from the performance. In this respect, updating the time required for execution could be quite straightforward; if the change in the average of time required is significant and maintained during a period of time, the value is replaced. Inferring the workload level from the performance would not be so simple, but it could be estimated from the actions of the pilot in the GCS. Whatever the method, when a significant change in the performance of a task or task overlap is detected, it should be propagated to all those cells in the workload profile containing overlaps where the value is involved. The update could happen due to the improvement of the skills of the pilot (a fact that is shown in [71]) or because the system had access to a real measure of the workload for a task or task overlap and is able to substitute the value estimated during the initialization. Same update applies to the workload threshold of the pilot. This could be done based on the performance; if there is a mantained improvement during demanding situations, the threshold could be slightly increased and the impact in the performance monitored in case there is a need to revert the change.

2.6 USABILITY AND AWARENESS

Although autonomy and automation are essential to allow a pilot to control several aircraft, the usability of the GCS is key to allowing a quick response and an appropriate awareness. [72] addresses a key aspect for the NtoM success: the information display and the control interface when pilots are responsible for multiple RPA. A GCS is a set of displays and inputs required to control the aircraft. Some are built with physical controls (knobs, levels, buttons), while others are mainly pieces of software and pilots basically use a mouse and a keyboard and maybe a joystick. For the specific case of the multi-RPAS, the Air Force Research Laboratory (U.S.) developed the Flexible Levels of Execution – Interface Technologies (FLEX-IT), where four different ways to control the RPA were evaluated to discover the one preferred by pilots considering that it should let them easily turn attention to other flights. That chosen was the so-called Noodle Control Mode. This mode allows the pilots to draw a line (the noodle) representing the flight path that the aircraft must follow. They can specify the noodle duration and altitude, airspeed and heading at the end of the path. The flight control algorithms take into account those values, the path, and the specific aircraft capabilities and constraints to calculate the commands and parameters required for the aircraft to follow the line reaching to the end of it with the specified values. This possibility of design of the autonomous actions releases the pilot from a constant dedication, who is able to control other RPAs while the programmed actions are being executed.

While the example of the Noodle Mode seems mainly addressed to surveillance missions in segregated airspace, which is not the target type of mission for NtoM, it is a good example to illustrate how the GCS usability and possibility of automation are basic pillars for safe, effective and productive multi-RPAS operations. As FLEX-IT shows, a high degree of automation and an intuitive and refined command interface and information display results in a decrease in the stress and workload of the pilot, therefore an increase of the number of flights each pilot can control simultaneously. The NtoM prototype assumes a high level of automation of the aircraft and the GCS; those configurations with greater requirements of attention and time from the pilot are not excluded, though, and would only determine the total number of concurrent flights per pilot.

NtoM does not include the GCS as a part of it and, while its usability would determine the whole performance and the quantity of flights that could be controlled concurrently, its development is out of the scope of the framework. Then, the efforts on the usability are focused on the software that the framework address to the pilots, which would be displayed in a dedicated monitor and has been conceived as a tool for the management of the flights: the NtoM Pilot Interface (NPI). This client also connects the pilot to the NtoM network. Despite not being exactly a GCS, guidelines for the design of GCS's [20] have been followed in the design of the NPI. [47] is an analysis of the human factors involving the RPA pilots and guidelines to avoid errors and mishaps. The document explains the weaknesses and critical procedures in the current GCS consoles. Those recommendations to reduce errors and improve usability were followed during the design and implementation of the NPI, although a final version would require a process of test, refinement and survey with real pilots.

Besides the advices regarding the GCS's procedures, a search was made in the literature about usability guidelines to follow in this type of applications. [55] describes an interface with common points with the NPI that follows Endsley's [73] guidelines for smart interfaces. These describe what kind of information should be displayed and the functionalities for a better user experience. Any implementation of the NPI should try to follow them too. The section about the prototype details these recommendations and how they were reflected in the design decisions for the prototype NPI.

2.6.1 The CPDLC Display

One of the main challenges for a seamless integration of the multi-RPAS operations is minimising any impact on the work of the ATC due to delays, oversights or errors caused by these operations. CPDLC provides a great basis to build tools addressed at avoiding those problems. As indicated in the assumptions, NtoM considers a mature implementation of CPDLC, and therefore leverages its potential.

CPDLC is a technology giving its first steps towards being the principal method of communications above current voice transmissions. By now, just a small subset of all the message elements envisioned are used, and the service is supported in a few areas and airports [74], while is being gradually implemented and learned by pilots and controllers. Probably the main drawback of CPDLC is the time required to compose and send the message in comparison with voice communication. Current CPDLC displays used in cockpits are far from intuitive and increase a lot the head down, something that should be avoided [[36], Part I, Appendix to Ch. 2, 6-f]. In fact, several recent proposals try to alleviate the response time and improve usability. Some current data link displays, like the Multi-Function Control and Display Unit (MCDU) (in Boeing 747-400, 757 and 767) and the Multifunction Display (MFD) (in Boeing 777) embed the message management in the same FMS display. Other option providing a separate screen for the messages can be found in the Datalink Control and Display Unit (DCDU) (in Airbus A319, A320, A321, A330 and A340) [75]. Taking a look at these interfaces and how they work, with a menu navigation very similar to the nineties Automatic Teller Machines, some of them even monochromatic (find a video example in [76] or a web simulator in [77]), it is easy to guess that even the less ambitious current computer graphic user interface can easily improve their usability.

The mechanical and ergonomic limitations in a cockpit have much less impact in a GCS, but there is still a delay in the response time implicit in the use of data link [47]. Current voice communications of unmanned aircraft add a delay that is mainly accepted by controllers when it is short [78]. While at the moment of writing this document no comparison was found regarding delay times of RPAS pilots vs. manned aircraft pilots when using CPDLC, [78] analysed the delay of unmanned aircraft when using voice communications and this can serve as a reference to some extent. If the communications of an RPAS pilot can suffer some delay due to the latency of the data link, and if that time can be increased because the pilot is attending another flight, then reducing as much as possible the time required to use the display is key to avoid more contributors to a delay that could hamper the work of the controllers.

Among all the components contributing to the end-to-end response time, NtoM can try to mitigate two of them. First, the equivalent to the pilot verbal communication delay, defined as *"The lag between the end of the ATCo clearance and the beginning of the pilot's readback"*[78]; in the present case the term *verbal* should be substituted by *data link*, and the beginning of the readback would correspond to pressing the button to send the reply. The second component would be the pilot execution delay, defined in the same reference as *"The lag between the end of the ATCo's command and when the pilot begins to initiate the manoeuvre"*, which in this case would end when the pilot sends the command, excluding the time required for the command to reach the aircraft. CPDLC allows the system to clearly parse the semantics of the messages and this can be leveraged to automate some actions, making the previous delays insignificant.

Implementing the message exchange following the requirements and flexibility specified in the ICAO guidelines about the procedures, the information displayed, the graphic and aural warnings, and embedding all this together with the NtoM needs while offering a good usability was a challenge, but the resulting display (Fig. 16, 4) seems to achieve these goals providing a minimum head down.

Current ATC tools allow CPDLC messages to be composed and sent by opening a dialogue from the same flight radar tag (see this in action in [79]). A similar approach has been suggested for the cockpit of manned aircraft [80], with a small pop-up box overlaying a flight deck situational awareness communication system when a message arrives; this menu would also allow the message to be composed and sent in a fashion very much like that of the ATC current tools. This option seems to be sufficient for simpler messages, and it has the benefit that the dialogue is so reduced that pilots do not need to move their sight to a side panel or monitor to compose the message. To this respect, the NtoM display would probably be preferred in the future if the potential of CPDLC is taken to its full advantage and data link dialogues become more complex, using the whole message set, multi-element messages or even interleaved dialogues. In any case, while these small dialogues seem acceptable from the ATC point of view, as an aircraft does not stay too much time in a sector and just a few messages will be exchanged, a dialogue providing such a constrained perspective of the communications would not be so desiderable for a pilot. Although [80] offers the possibility to navigate through the message history, this is done as in a common MCDU, with a high head-down. However, one of its virtues is that it displays on a map the current planned flight route and a preview of the instructions suggested by the controller, as well as the information required to make an informed decision about the reply. This allows a quick reply with no need to move attention to different sources of information. It also adapts its behaviour to anticipate the pilot's actions: if the pilot selects WILCO, it offers the possibility to load and execute the actions, which would automatically send a WILCO to the ATC; if the reply is UNABLE, it offers the composition of the DUE TO message element. The NtoM display coincides with it in the first two features: a button to execute the instruction and the automated reply when used.

Another suggested approach for a more comfortable composition of the messages from the cockpit [81] is a kind of extension for the existing different displays the pilot interacts with (Primary Flight Display (PFD), Electronic Flight Bag (EFB), etc.). According to this, clicking a value on any of those displays would open a contextual menu to select message elements in which that value is involved. If any of the options is selected, a composition screen appears, and can be used to compose and send the message. This feature does not exclude the possibility of having a dedicated CPDLC display, or having the message composition dialogue embedded in the FMS; then, it seems an interesting add-on to quickly compose simpler messages as long as it could be installed in the mentioned displays, it had a display able to show the message log and could be able to update the log in that display when messages are sent from any other screen. The goal of this composition shortcut was to prevent the pilot from needing to move to the communications display and copy information appearing on another display, thus reducing the head-down and the possibility of input errors. A similar solution is that of [82], which suggests a CPDLC context manager; considering different independently CPDLC-enabled avionics devices; this manager would synchronise them all to compose or reply to the messages. For instance, activating the CPDLC mode in the context manager, the subscribed devices (those participating in the CPDLC coordination) enter into a composition mode and the pilot could set the altitude value of a message element by using the altitude knob in the glare shield, then selecting a waypoint in a display and finally reviewing and sending the message from the MCDU. For received messages, activating the CPDLC mode would allow replies to be made from those devices from which an input would make sense according to the received message, and these would suggest message elements or values for the composition.

Finally, other proposals are aimed at assisting in the composition of particularly error-prone messages, or those with a higher head-down. In [83], previous to requesting a clearance to change a flight parameter like altitude, speed or a route deviation, the pilot introduces the desired change in the FMS. Using external data sources and considering the criteria for optimisation, the FMS calculates and suggests the values for the rest of the parameters that should be changed and therefore requested to the ATC. Instead of manually entering those values in the MCDU, in this system, the message is populated automatically with the data from the FMS. Similarly, [84] allows a quick composition of the messages to request In-Trail Procedures (ITP) altitude change.

All these previous suggestions illustrate the need for a quick, comfortable and error-free composition. These issues were taken into account during the design of the display for the prototype. The resulting implementation is described in the section describing it later.

2.6.2 Control Migration

Transfers of control can be scheduled tasks due to workforce rotation, coverage, the result of a contingency measure, or requested by the pilot or the supervisor for different reasons. A flight handover is a delicate, slow, cognitively expensive process prone to errors [85] that should be avoided as much as possible, for instance avoiding an assignment to a pilot that is ending his/her workday, which would force another transfer. It has been a source of errors with dramatic consequences when the procedure required manual configuration or the GCS would not clearly show the status of the process [18]. Inspired by the lessons learned from the accidents during the handover process in the military environment, NtoM suggests an orderly and informed procedure which, orchestrated by the system, does not leave a chance to GCS misconfiguration incidents. The system only allows the aircraft to receive orders from a single pilot at once; when the transferring pilot participates in the transfer, the receiving pilot decides when he or she has sufficient information and awareness to take the control and is the transferring pilot who effectively executes the transfer. Find in the section about the prototype how exactly was implemented this process and a screencast of it.

2.7 MONITORINGS

The monitoring of the action of the pilot is important when they control several flights, as they could mix intentions, as was shown during the experiments. It also serves as a crosscheck in what will probably be single pilot operations. While CPDLC provides unambiguous and persistent communication, the cross-verification by both pilots before the execution of the clearances has been recommended in manned aircraft same as is done with voice communications [86]; in this sense, the NtoM monitoring would play the role of the first officer. An example of this need using CPDLC is reported in [87], when pilots do not realise the condition of the conditional clearances and execute the instruction immediately. There was also a problem with the terms "AT" and "BY", which resulted in different interpretations of the conditions [88]. These problems have led to a rewriting of some of the message elements and the improvement of procedures. Anyway, the GCS Monitor, able to unequivocally interpret the semantics of the messages, including those with a condition that can be tracked, like position or time, is able to check if the pilot is executing what is required when is required and send a warning or confirmation request if not. Where CPDLC is not available, an option could be the use of reliable speech recognition for the system to parse the meaning behind the speech. [89] is a recent work that shows the feasibility of this option, and [90] is a proposal for detecting non-compliance actions to both voice and data link messages instructions.

Being the NtoM system able to understand the communications, and having access to the flight definition, GCS activity and aircraft telemetry, it can monitor if it is evolving as expected and act consequently if not, like triggering warnings or emergency procedures. But the system can also send reminders, which is a particularly interesting safety measure when controlling several aircraft, which increases the possibility of forgetting a task, in particular conditional instructions that should be applied at a time or position ahead. Following are the monitorings envisioned. Some of those implemented in the prototype can be seen in action in [43, Screencast 8].

2.7.1 Unexpected Actions and Breach of Orders

Regarding compliance with the ATC instructions, the use of CPDLC instead of voice allows to extract the semantics of the message without amgibuity and monitor its execution, as recommended in [36], Part I, Ch. 2, 5.9: *"Monitoring automation capable of questioning certain classes of operator actions that can potentially compromise safety must be designed into the system"*. So, if the system detects that these are ignored or contradicted, after warning the pilot and supervisor, the control of the aircraft could be automatically transferred, with no need for approval from the current pilot.

The system will detect and react if the pilot tries to execute an action for which an ATC clearance is previously required, like a change of flight level, or one that contradicts that specified in the flight plan. This includes the case when an instruction from the ATC has been accepted but then not executed, or not correctly done. Some ATSUs do already have conflict detection and conformance-monitoring functions based on Automatic Dependent Surveillance and/or secondary surveillance radar (SSR) data, but in NtoM the system would check the commands conformance just when they are applied and could alert controller and operator of any disagreement, providing a more quick and informed picture of the situation. It could even automate a contingency measure to avoid dangerous situations until the situation is clarified and solved. This kind of monitoring is not assuming a lack of diligence from the pilot, but the fact that controlling several aircraft concurrently could lead to mix intentions, especially during moments of stress. Ultimately, the system acts as a kind of silent FO constantly checking the actions and only warning about unexpected situations if these appear, providing a safety check to avoid malicious actions of hacked connections.

A first measure when any of these situations are detected could be to send a warning to the pilot because maybe it was only an error. If the action is not corrected, the supervisor could also be warned and he/she could use the chat to communicate with the pilot and clarify the reasons behind. If the pilot does not solve the problem, a deliberate action can be associated with a reckless behaviour or a pilot indisposition and the system could automatically force the transfer of the flight(s) of that pilot. It is possible in a future that the policies behind a situation like this will be subject to safety regulations instead of leaving such decision in the hands of the airline. These policies could reply to questions like when to take away a pilot from the control of the aircraft, because this decision could have consequiences on the final responsibilities in case damages are involved. Fig. 9 describes a suggested procedure when a pilot exhausts the time expected to execute an instruction from the ATC. First, it would only send a reminder to the pilot. After a margin of time if the task has not been executed yet, if there is a supervisor available, he/she would receive the warning and the system would let to the supervisor the responsibility to clarify and solve the problem. If there is not supervisor available, the pilot would receive a more striking message, noting that, if the task is not completed, the control will be transferred, which would happen a period of time after this warning. Forcing the transfer of a flight should be a last resort; it is disruptive for the receiving pilot, and a slow and delicate procedure that could affect the traffic and the work of the controllers. That is why the system

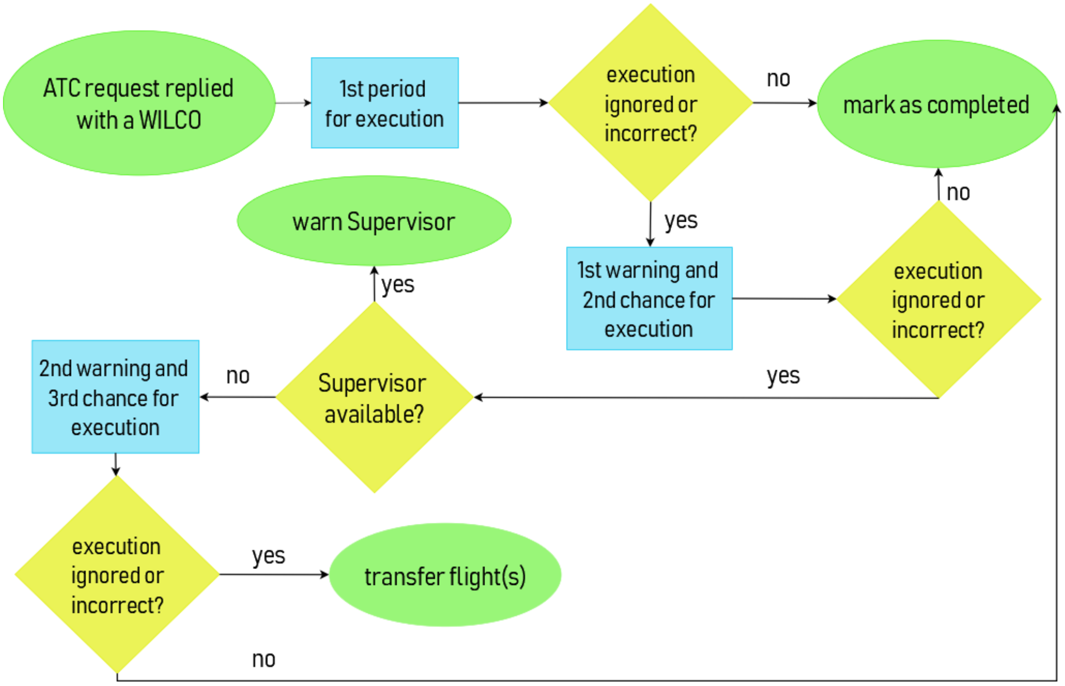


Figure 9: A suggestion of procedure for the case of a task not being executed.

offers the pilot two or three opportunities before applying it. Similar procedures should be decided for each kind of contingency.

About how to warn controllers of this kind of situations, two types of monitorings and warnings can be differentiated. One of them, which can be found in common ATM systems, monitors the behaviour of the aircraft and warn controllers when the trajectory of the aircraft becomes different from that expected. Controllers are warned of this with a code in the radar tag. A second, still not widely implemented option, is to detect the change from the very moment the pilot applies the command in the aircraft. This can be done using ADS-C which, with its provision of Extended Projected Profile (EPP) allows sending a report to the controller when there is a modification of the flight plan or any parameter, providing details like ground vector, air vector or short-term intent. The NtoM system tracks the development of the flight like this second method by default, and once aware of the unexpected situation, could automate free text CPDLC messages to back up these reports with more information, like the possible cause, or substitute them when the 4D trajectory capability is not available. A broadcast of the intent to the nearby traffic could also be a useful option.

2.7.2 Link Problems

The monitoring can also be applied to the link quality. When detecting a dangerous link degradation that could affect the service, the system could warn, and suggest or assign, if possible, an alternative pilot with better coverage. Consider that, for NtoM, pilots could be scattered around the world.

In case of an LLP activation, the autonomous report would be that specified in the pre-loaded procedure, or a ADS-C notification to the CDA if possible (if the link is lost with the GCS but not with the ATSU), specifying the emergency status variable of these reports. The behaviour when this situation occur is expected to be described in the flight plan. ICAO's RPAS ConOps [91] specifies that the RPAS flight plans must comply the same conditions and contain the same information as those for manned aircraft. But, besides, ICAO's Annex 2 [92] leaves room to extra fields and information like *3.3.3.1 Whatever the purpose for which it is submitted, a flight plan shall contain information, as applicable, on relevant items up to and including "Alternate aerodrome(s)" regarding the whole route or the portion thereof for which the flight plan is submitted*. From this statement, can be understood that the envisioned behaviour of the aircraft when facing a lost link or any other contingency, should be indicated for the whole trajectory, and we could think of different procedures or alternative aerodromes depending on the area and the contingency. Currently, some countries require the RPA to transmit a transponder code — 7400 in the U.S. and Australia — when this happens, and controllers should have information in the documentation about the LLP that would be loaded. But, handling different procedures, if the controller has no access to the events determining the LLP chosen, he/she could not guess the one being executed and the future behaviour of the aircraft (the navigational intent can be retrieved via ADS-C) if the pilot is not able to communicate. Including an automated free text CPDLC message during the execution of the LLP with the identification of the plan being executed would be an added safety measure. If the link with the CDA is not possible, the NtoM system could back up the notification, if there is an alternative ground way of communication with the controllers, providing the expected behaviour given the last readings available from the aircraft and the preloaded procedures.

The question of providing the controllers with the best awareness possible during non-nominal situations was addressed in the work that was the origin of RAISE, the simulation environment used in the experiments of this thesis. RAISE was developed by ICARUS research group to evaluate the impact of the RPASs integration on the ATM as part of the WP-E project ERAINT (Work Package for exploratory research activities on Evaluation of RPAS-ATM Interaction in Non-Segregated Airspace) [93]. It was used to evaluate how different kinds of contingencies, where the aircraft was left with different capabilities, affected the work of the controllers (different combinations of engine failure/link loss, able/unable to provide flight intent, with/without pro-active autonomy). The experiments highlighted the need to provide the ATC with all possible situational awareness considering the limitations of the RPAS as a result of the problem that lead to the non-nominal situation. To satisfy that need, the ConOps suggested to address those situations, exploited the use of ADS-B and ADS-C.

2.7.3 Pilot status

Another projected monitoring is the health status of the pilot, a feature also proposed, even more justifiably, in SPOs [37] [94]. The implementation would be quite straightforward for remote pilots, with each interaction in the interface acting as a deadman pedal; when the low activity does not provide enough interaction, the client would open a dialogue requesting for feedback. A more sophisticated approach would be to use biometric devices, as suggested in NASA's SPO [94],

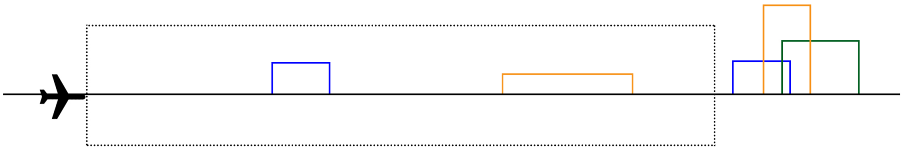


Figure 10: Workload forecast window. Conflicting overlaps will not try to be solved by the system until they do not enter in the time window being monitored.

which also considered the possibility of a video feed of the cockpit to check the pilot's behavioural state and clearly determine any possible pilot incapacitation and the need to transfer control to ground operators or the onboard automation. This approach also tried to provide the ground FO with the information provided by the body language when both pilots share the cockpit. In NtoM, the behaviour of the system facing a lack of feedback could be a first showy and noisy warning to the pilot; if still not replied, a warning to the supervisor, who would try to get in contact with the pilot; if the contact is unsuccessful or there is no supervisor available, the pilot's flight should be transferred.

2.7.4 Workload Monitor

This feature mixes two categories of measures: the workload management and the monitoring by the system. Like IMPRINT does, NtoM keeps a constantly updated graph of workload forecast for the remaining trajectory. From the events and the readings, the workload timeline is dynamically updated to add, delete or move the expected peaks. In IMPRINT this is provided to help the system designer to detect potential problems; when a resources overlap incompatibility appears during the simulation, this will mean an increase in the time estimated for the task to be performed or the impossibility to carry out with it. In NtoM the timeline forecast does not represent actions in the GCS but tasks in the different flights, which will help to identify unbearable levels of workload or task overlap incompatibilities. The maximum level of workload that each pilot can handle will be determined from the pilot's workload profile. Anyway, the total workload should be kept under a safe margin from the pilot's top limit for the pilot to have a chance to control the situation when facing any unpredictable and sudden increase.

When excess of workload or overlap incompatibility is expected, the system will evaluate the situation to decide the measures to take. In this respect, to avoid the unnecessary application of measures, it seems appropriate to consider a time window of forecast instead of the whole remaining trajectory. Fig. 10 depicts this idea. As can be seen in [43] Screencast 1 and 2, the evolution of the flight — changes of speed, deviations — has an impact on the overlap of tasks. Problematic overlaps ahead on time could resolve themselves with no need for intervention from the system, which could apply disruptive measures. Therefore, the time window would limit the monitoring of these overlaps but its width should provide a sufficient margin of reaction for the system and the pilot in case of need.

The timeline forecast would also be updated with the current workload perceived by the pilot, which could be measured using biometrics (physiological parameters like heart rate or pupil dilatation [65]) or any other non-invasive method that could hamper the operations. If these sophisticated elements are not an option, it could be estimated comparing the current task(s) performance with the values of the workload profile.

MEASURES TO SOLVE EXCESSIVE CONCENTRATION OF WORKLOAD The evaluation of strategies for the mitigation of workload contingencies opens its own branch of research. A first response of the system, could be to try to reduce it somewhat. How to do it would depend largely on the kind of tasks involved and if there is a possibility to take control of some of them. Considering that the transfer of flights should be a last resort, a solution could come from the coordination of the flights involved [15], for instance with a change of speed in any of them to avoid the coincidence of tasks, or assisting the pilot somehow to alleviate the usual workload level or time required to execute the actions.

If the task overlap could not be predicted, suddenly appearing with no time for the system to mitigate it, the NPI, coordinated with the server, could try to provide some aids to the pilot. For instance, if a pilot with a high level of workload receives an ATC request and the aircraft timeout (the time to reply) is about to expire, a synthesised voice could ask the pilot if a STANDBY should be sent. Using voice recognition a "yes" would be enough to gain some extra time and the pilot does not need to swap to that flight to attend the message, which would increase the cognitive load. Similarly, the client could read aloud the waiting message and ask the pilot if the instruction can be replied and applied and do it if the answer is positive.

Ultimately, if the system detects that the pilot is surpassed by the situation, a handover should be suggested or automatically executed by the system. The system should evaluate the situation because a transfer of control when a high peak of workload is detected could be more risky than just letting the pilot deal with this high workload if it is only expected to last a short time. When a flight must be necessarily transferred but all pilots show incompatibilities at some point of their timelines, the system should evaluate and guess the less critical situation. One option would be choosing the pilot with the farthest conflicting point, to facilitate that the problem could eventually disappear when the moment arrives, considering that the development of the flight will probably move the expected peaks of workload. Other option would be to select that with an overlap of tasks that can be supported by the system, releasing the pilot from them.

This whole idea of the evaluation of the current and future status of the aircraft and workload of the pilot to determine the response of the system can be found at a task level in [55], a suggestion for a pilot/vehicle adaptive cockpit interface focused on the rather more demanding needs of the combat pilot, but with a final aim in line with the requirements of NtoM concurrent piloting: *"improve the pilot's situation awareness and decision-making while alleviating workload"*. On it, the workload evaluation will be used to select the kind of information provided to the pilot to perform the task and balance the stimulus modality (visual, auditory, haptic). [55] takes into account the impact of the task switch in the situation awareness, similarly as NtoM needs to consider this impact when the switch is made between flights too. It also advocates the need to reflect the individual preferences and limitations to cope with the workload, which could evolve as experience is gained, something that would also be reflected in the NtoM pilot workload profile.

An added function of the workload monitoring is that pairing the record of activity with the performance-based evaluations and/or physiological measurements would allow to update the pilot's workload profile to reflect the improvement of skills with experience.

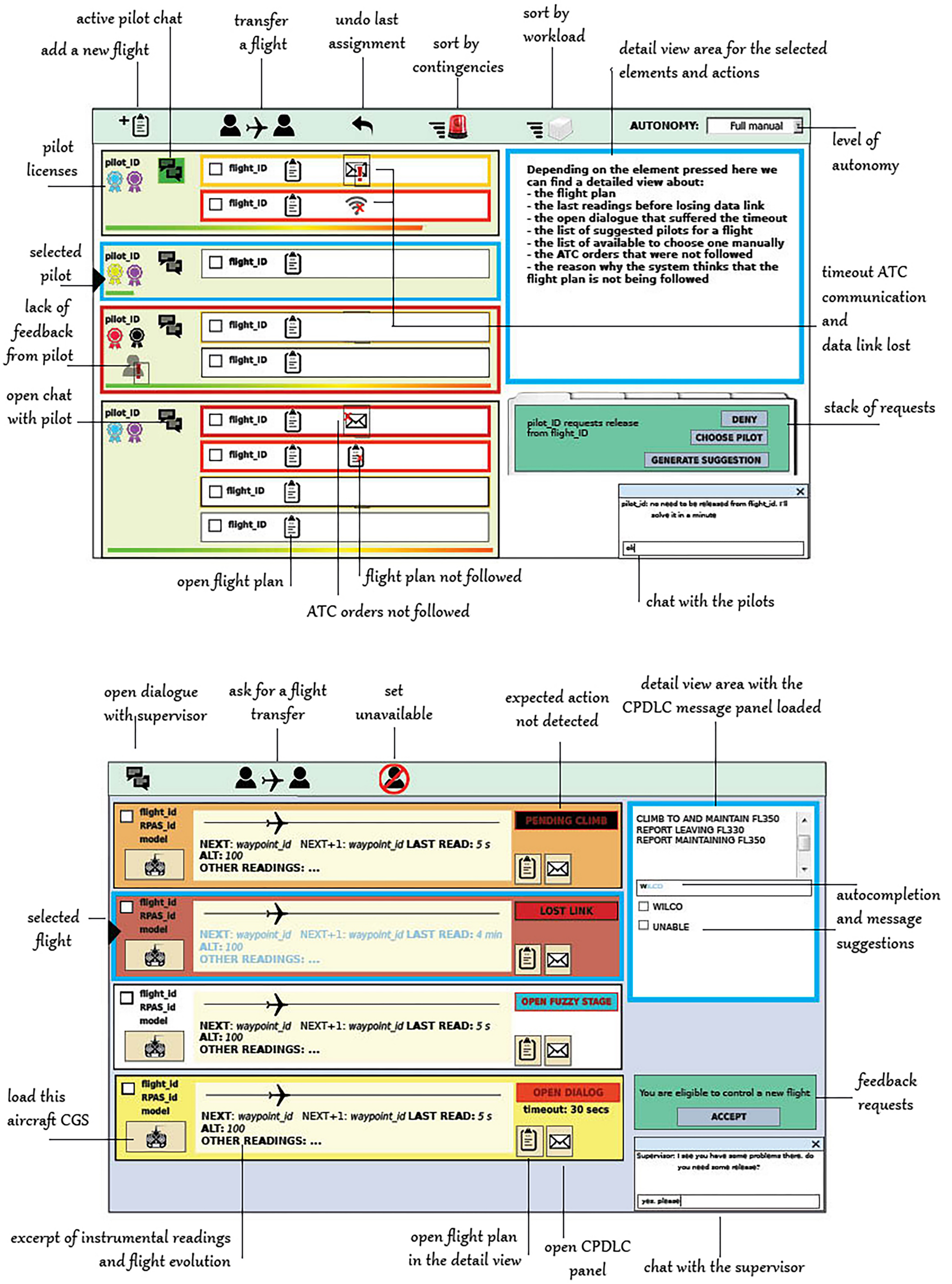
3 PROTOTYPE

The prototype was built to test and illustrate the proposal, and it was useful to discover other needs and inspire new aids. This section describes the implementation, features and design decisions included in the prototype to support the hypothesis of the research. Due to time and resources limitations, only part of the features, tools and components suggested in the thesis could be implemented, and most of them can be seen in the screencasts indicated in each case. The prototype includes three different clients for controller, pilot and supervisor roles. The client for the controller is not part of the framework but was added to the prototype to provide a tool to simulate the ATC interaction during the experiments. This role was automated later, and the messages sent from the controllers can be scheduled in a script too. An operative implementation of these interfaces would require further iterations, interviews and testing with final users. Fig. 11 shows the initial sketches for the pilot and supervisor interfaces.

Following descriptions are focused on the pilot client and the server, which were the pieces more advanced in the implementation. Current supervisor client implements only basic functionality: loading and pre-flight scheduling of flights, a chatroom with pilots, notifications and the print of each pilot workload forecast; then, this interface will not be seen in detail, although it appears in some screencasts.

3.1 SYSTEM ARCHITECTURE

The architecture of the prototype is different to that explained in the previous chapter that would correspond to a real implementation. Now, the GCS is a kind of terminal without direct connectivity with the aircraft and is the NtoM server the final responsible for sending the order to the instance of the aircraft in the simulator (Fig. 12). This seems an unlikely configuration for real implementation, as it would add the network delay to the one of the C3 link. In the prototype, the server reads the aircraft's simulated Automatic Dependent Surveillance-Broadcast (ADS-B) readings from the socket of a single instance of RAISE. The RAISE simulation environment mixes the Instrument Flight Rules (IFR) traffic provided by the eDEP [95] simulator and its own simulations of RPAS. RAISE broadcasts ADS-B-like outputs with the aircraft readings through a socket, concretely: callsign, flight level, ground speed, heading, latitude, longitude, vertical rate and timestamp. These are read by the server, which sends them to the pilot and controller clients to update their values. In the opposite direction, the server receives the control commands from the NPI and sends them to RAISE. To allow this, a basic interface was coded on the eDEP simulator used by RAISE to read the commands from a socket. The



Depending on the element pressed here we can find a detailed view about:
- the flight plan
- the last readings before losing data link
- the open dialogue that suffered the timeout
- the list of suggested pilots for a flight
- the list of available to choose one manually
- the ATC orders that were not followed
- the reason why the system thinks that the flight plan is not being followed

pilot_ID requests release from flight_ID

pilot_id: no need to be released from flight_id. I'll solve it in a minute

ok

CLIMB TO AND MAINTAIN FL350
REPORT LEAVING FL330
REPORT MAINTAINING FL350

☐ WILCO
☐ UNABLE

You are eligible to control a new flight

Supervisor: I see you have some problems there. do you need some release?

yes. please

Figure 11: Initial sketches of the supervisor (top) and pilot interfaces.

limitations of eDEP only allowed a few instructions like changes of speed, level, heading and direct to waypoint to be loaded.

The benefit of this configuration appears in the use of the RTI Connext connectivity framework [96], compliant with the Data Distribution Service (DDS) stan-

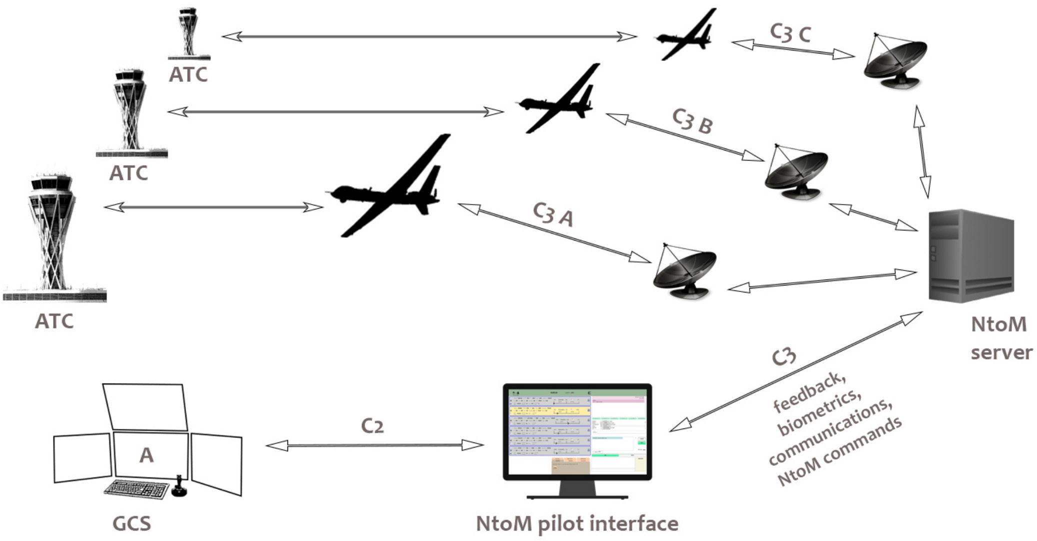


Figure 12: The architecture of the software components underlying the prototype.

dard, to communicate between server and clients. This middleware allows the definition of publishers and subscribers attending specific types of messages. Fig. 13 shows these elements and their relationships in the prototype for the case of the NPI client. There are two different message datatypes: one for the CPDLC messages, handled by the CPDLC Service, and a second with a more general purpose, like commands, readings or intra-network communications, handled by the Commands Service. In the figure, round items (circles and ellipses) represent topics, which are the nexus between publishers and subscribers. A Publisher can contain several DataWriters, which is the object that, ultimately publishes the messages; these DataWriters specify in their messages to which topic are addressed. Similarly, a Subscriber can hold several DataReaders, each of them listening to messages of a particular topic. When a DataWriter writes for a topic, it will trigger the callback of the DataReaders subscribed to that topic. In the prototype, clients send Command messages to the server by publishing messages for a topic to which a DataReader in the server is subscribed. Similarly for CPDLC messages. However, each client has its own topic to receive the messages of each datatype and the server dynamically creates their DataWriters as users log in the network.

The decision of using two different datatypes is based on modularity. For instance, in the specific case of the supervisor client, it does not use CPDLC messages. Also, if the CPDLC display wants to be decoupled and used as a standalone application, there is no need to include all the part regarding the commands datatype and its service in the server. Also, the middleware allows straightforward deserialization of the messages for the custom datatypes, which are defined in IDL (Interface Description Language) files. Fig. 14 shows the example of the fields in the IDL definition for the CPDLC messages datatype. The middleware allows also a simulation of different scenarios of Quality of Service (QoS), providing a framework to test contingencies related to link quality to *"determine what compensatory behaviours, if any, air traffic controllers and UAV operators adopt in response to*

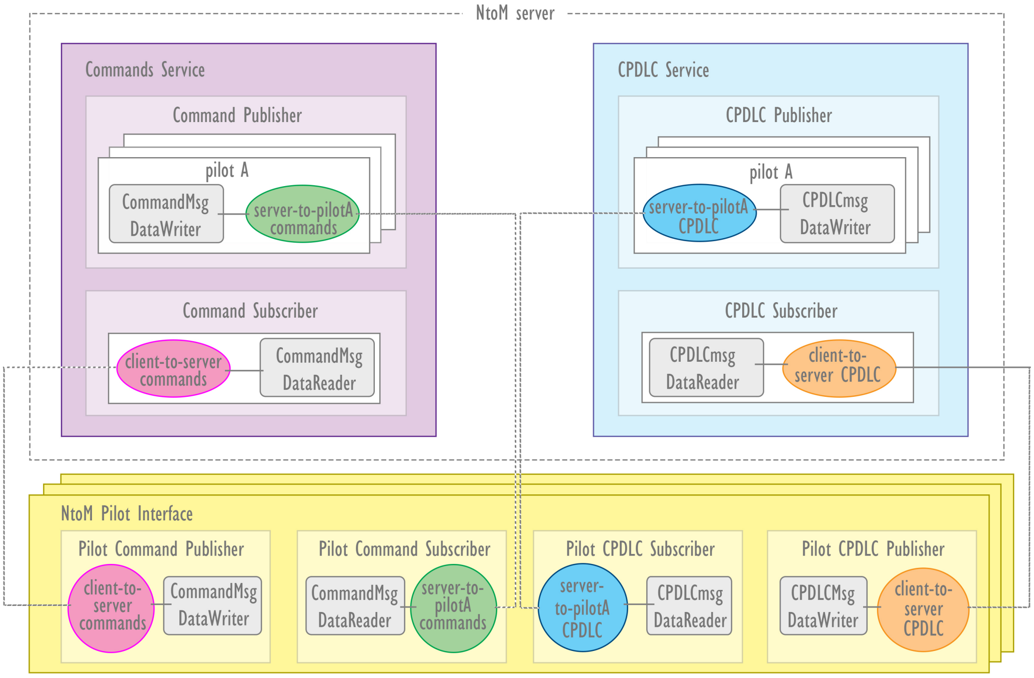


Figure 13: DDS connectivity of the prototype.

communications delays" [97] or provide training in the associated established procedures.

In the case of the prototype, the server acts as a router for the messages between controller and pilot; when a message arrives from any of them, the CPDLC Service asks the GCS Manager which pilot or controller should receive the message given the callsign (as pilots and Current Data Authority can change along the flight). If the message comes from the ATC and it includes instructions, the GCS Monitor creates what has been called an Open Order, which represents the actions that should be followed by the pilot. This Open Order is not still active and monitored, that will only happen if the pilot replies with a WILCO to the message. An UNABLE would delete the Open Order.

3.2 NTOM SERVER

Fig. 15 shows the components of the NtoM server prototype and their relationships:

EVENT HANDLER This component applies the policies established when a situation requires a decision or an action from the system. Acting somewhat like the so-called *decision authority* in adaptive automation, it will trigger notifications, requests for feedback or automated actions. The requests for management are sent by the Workload Monitor or the GCS Monitor. When it happens, the Event Han-

```
const long MAX_ELEMENTS = 7; /* Maximum Elements per message */
const long MAX_VARIABLES = 5; /* Max. variables per Message Element */

/* Parameters for a Message Element. */
struct MessageElementVariables {
   sequence <string, MAX_VARIABLES> variables;
};

struct CPDLCmsg{
     string aircraft;     /* Callsign */
     Flow flow;           /* Downlink or uplink */
     unsigned short MIN; /* Message Identification Number */
     unsigned short MRN; /* Message Reference Number */

     /* List of Message Elements' codes */
     sequence <string, MAX_ELEMENTS> identifiers;

     /* List of lists for the variables for each Message Element */
     sequence <MessageElementVariables, MAX_ELEMENTS> parameters;

     long elapsed_secs;  /* Timestamp of reception */
};
```

Figure 14: IDL Data Type definition of the values in a CPDLC message.

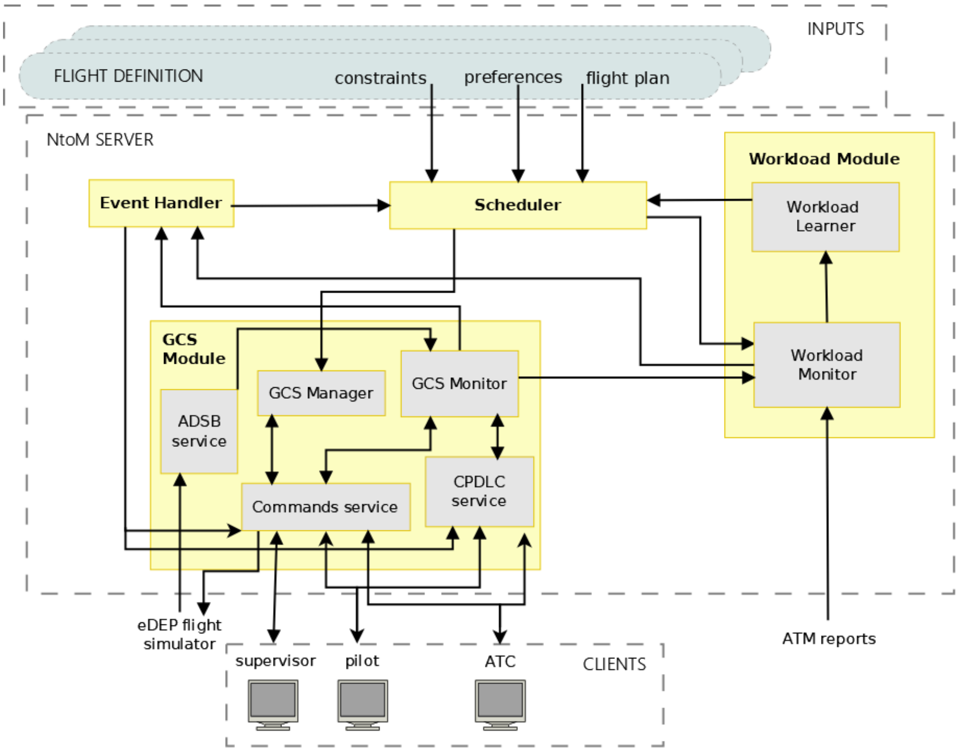


Figure 15: NtoM server modules.

dler will evaluate the problem with the information available and the options to solve it, with actions that could include sending warnings or automating tasks.

SCHEDULER Its mission will be to find the optimal distribution of flights among pilots taking into account all constraints and preferences. This can be done before or during the flight, and can be directly applied or not depending on the automation mode chosen by the supervisor.

Loading a flight to be scheduled implies reading an XML file with the flight plan and other NtoM specific parameters, and convert it to the format understood by the flight simulator used by the prototype. The initial ATC-flight assignments are also loaded from an XML input.

THE GCS MODULE

- *GCS Manager*. The component where all the entities (flights, pilots, controllers) and their relationships are represented and managed.

- *GCS Monitor*. Having access to the information provided by the GCS (actions of the pilot, aircraft status and telemetry, ATC communications), this component keeps track of the evolution of the flight to determine flight plan and ATC instructions compliance. When it detects a situation that requires an action from the system to solve or avoid a problem, it requests the Event Handler to attend it.

- *ADS-B Service*. This piece of the system listens to the aircraft readings broadcasted by the flight simulator and sends them to the GCS Monitor, which will also forwards them to update the clients.

- *Commands Service*. Responsible for sending and listening for the reception of messages used by the framework: pilot actions, flight assignments, notifications, chat messages, etc.

- *CPDLC Service*. Similar to the Commands Service but specific of the CPDLC messages.

WORKLOAD MODULE Contains the components involved in the workload management.

- *Workload Monitor*

  Using the sources of information available, it builds the timeline of tasks along the flight. The resources required for each task will be retrieved from the Workload Learner, the component responsible for maintaining the workload profile of the pilot. The resulting timeline will be used by the Scheduler to check if the assignment of flights is admissible.

  The Workload Monitor is responsible also for tracking the real workload and stress of the pilot. The real workload could be different from that estimated in the forecast due to events like meteorological conditions, non-nominal situations or ATC requests adding unexpected tasks. The stress of the pilot could be monitored by physiological measurements if wearables are available for pilots or inferred from the pilot performance (delays, errors); these data would be provided by the GCS Monitor. If this real workload or stress surpasses a threshold, the Workload Monitor will warn the Event Handler, which will evaluate the situation and apply the required measures. In the prototype, the workload forecast is updated every 10 s.

- *Workload Learner*
  As the aircraft overflies a waypoint associated with a sample of the workload heatmap, this component updates the sample adding the real experience of stress of the pilot. Proceeds similarly for the workload profile of the pilot, overwriting the resources assigned to the estimated task overlaps or updating the values for the tasks as the pilot gains experience or the task requirements change.

DATABASE Current prototype does not include a database to store the activity of the platform to provide future diagnostics and analysis. However, the log of the actions of the pilots was required to analyse the response time and errors, so these are stored in XML files. Initially, it was considered to store the flight events in a database using the Flight Object structure. The Eurocae's Flight Object concept [98], a future mandatory standard, is a complex and detailed representation of the flight plan and events during the trajectory conceived to be shared among ATSUs. Then, there is not a direct need to include this structure in the concept and the motivation was mainly using an existing representation. But its complexity made it quite costly to implement and handle and its level of detail is quite higher than the one required by the system, which made it not suitable for a structure that requires to be constantly acceded. Also, the available documentation raised many doubts about its correct use and the meaning of the fields and there was no subject matter expert guidance available to check the correctness of its interpretation. Then, while initially a simpler object, a Flight Object subtree, was implemented, its use was discontinued during the development of the prototype as its contribution was not clear and its complexity was ballasting the implementation.

3.3 THE NTOM PILOT INTERFACE

As aforementioned, the design of the client for the pilot tries to follow Endsley's [73] usability guidelines. First, these say that the interface should provide a big picture of the situation, with high-level information, which could be the definition of the flight strips that represent each of the aircrafts. Second, the pilot should be able to decide which information is displayed. In this respect, the flight loaded in the GCS would not necessarily be the one active in the NPI; the pilot could control one aircraft in the GCS and reply to the ATC of a different flight, or apply the set of commands associated to the instruction received in another flight. Flight strips can be rearranged by drag and drop, and in the original mockup, it was planned to place non-exclusive order preferences, like *closer to next waypoint first*, *pending actions first* or *urgent actions first*. This would be optional, as some pilots could prefer to always find the same flight in the same place. Third, Endsley stresses the importance of providing information about past events that could help pilots to form an idea of the current status. In the NPI a pilot receiving a flight would have access to the CPDLC history, pending events, list of previous pilots and their annotations, or to an automated event logging. Lastly, there is the importance of providing a personalised experience, to consider each user preferences and limitations. The preferences could be satisfied by providing some flexibility about how and which reminders and warnings are triggered but with some limitations, like critical warnings that could not be turned off. Regarding fitting pilot limitations, that is the purpose of the use of the pilot workload profile.

The following subsections describe the implemented features in the prototype NPI. Due to time limitations, many of the envisioned functionalities could not be included, and some of those coded cover only some cases, to provide at least some examples of its potential. Some limitations guided the design decisions. For instance, the experiments would be performed by non-professional pilots during a few sessions; then, the use was simplified to avoid too much training and allow that the experiments could arise the difficulties of the multi-tasking rather than the difficulties of performing the tasks. A second limitation was not having access to a real GCS and professional pilots with whom to work side by side in the design and get an idea of the information they are provided with or the usual procedures and check lists, despite some collaboration was obtained by a subject matter expert and military RPAS pilot to clarify and highlight the main needs of these operations.

3.3.1 Flight Strips

Each flight assigned to the pilot is shown as an horizontal strip with the main readings providing a summary of the status of the aircraft (Fig. 16, 1). This allows the pilot to make a picture of the situation with a quick glance. In a real implementation, with a GCS plugged to the NPI, a button placed in each strip would allow to choose which aircraft is going to be controlled in the GCS, but in the current prototype the GCS was reduced to a set of commands embedded in the same NPI, which was enough considering the kind of experiments and the fact that the participants were not professional pilots. While in a real implementation the strip selected would not necessarily be that of the aircraft loaded in the GCS, in the prototype the control commands apply to the selected flight and therefore there is no button in the strip. When a strip is selected, the CPDLC display shows the history of communications of that flight and the messages sent will be addressed to its Current Data Authority (CDA), the ATSU assisting the flight at that moment. In concordance, the display will only print the messages received by the selected flight.

It must be stressed, in case the reader is tempted to perceive the NPI as an oversimplification of a GCS, that the NPI is envisioned as a support for the parallel piloting and does not try to substitute the GCS, which would be plugged to it, and should provide all the necessary information, functionalities and awareness. However, depending on the design and implementation of the GCS, it could be interesting to provide information and shortcuts to execution of commands in the NPI avoiding the use of the GCS. For instance, think of a GCS that, despite being able to keep several links, is only able to load the information and controls of one aircraft at a time. In a situation like that, the loading of a flight in the GCS could be relatively slow, and it would be possible that the action required is so straightforward that there is no need to load all the modules and displays of the GCS for that flight. Therefore, the number of times that the load is required in the GCS tries to be minimised by the summary of telemetry, the automation of tasks and providing shortcuts to send commands and communications to flights in the background (any flight not currently loaded in the GCS). To that extent, the readings included in the strips were the following (Fig. 17):

- *Current Link*. Whether command and control (C2) or command, control and communications (C3) datalink, this icon displays the mode (terrestrial, satellite) and its quality. In a real implementation probably some metric about the QoS would be used, but for the experiments it was simplified as a notion of signal strength and displayed with a typical signal bar icon. On satellite, the name of the band is displayed beside the quality. When using directional antennas, there is a quality icon both for the primary and the secondary links and their respective frequencies beside them (Fig. 17, left end).
- *Callsign*. Unique identifier for the aircraft.
- *Model*. Aircraft model.
- *Or.*. Waypoint of departure.
- *Dest.*. Destination waypoint.
- *Departs*. Time of departure.
- *FL*. Current flight level.
- *CFL*. Cleared flight level. When the pilot receives a clearance or an instruction to change level and it is accepted, the new level is printed here.
- *RFL*. The Requested Flight Level is the one requested by the pilot, seeking to optimize some aspect of the flight, like fuel consumption. The traffic determines its availability, and it may not be granted by the ATC.
- *Heading*. Compass direction pointed by the aircraft nose.
- *GS*. Ground Speed, as called for an aircraft the speed over the ground, the one taking the ground distance travelled to calculate the value. Measured in knots.
- *Mach*. Represents the speed taking as reference the speed of sound (when Mach is 1).
- *Last read*. Timestamp of the last reading, which is key in the unmanned case.
- *Lost Link Timeout (LLT)*. If the aircraft loses the link with the GCS, after the number of seconds indicated by this value, the aircraft will load and execute the procedure associated. This values can be changed during the flight. For

instance, we should not allow an aircraft out of control in areas of high traffic density; in such situations, this value would be of just a few seconds, while in lonely areas a timeout of minutes could be tolerated.

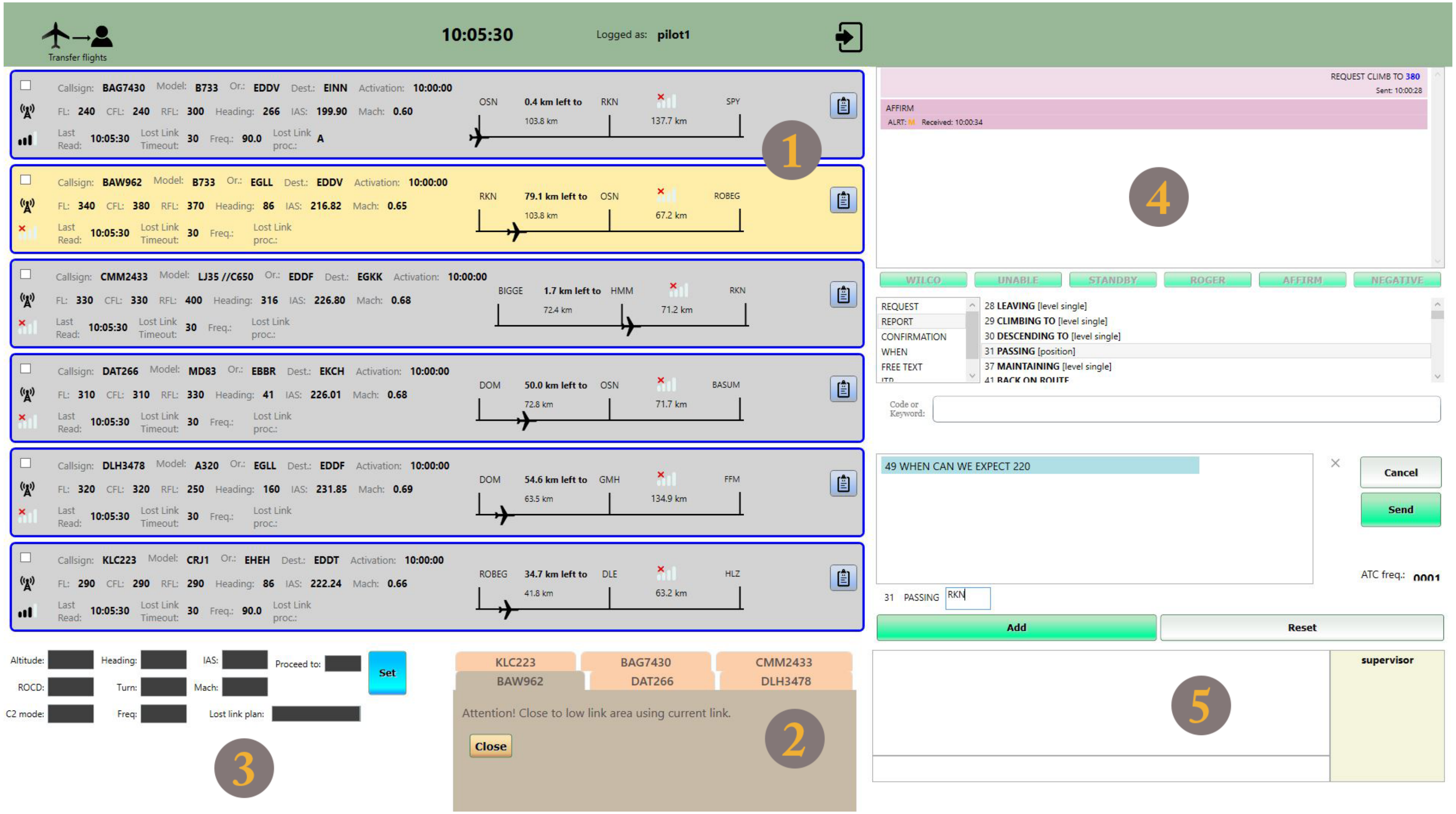
- *Lost Link procedure (LLP)*. The current procedure associated with the loss of link. Its value can change along the flight. An example would be a procedure landing the aircraft; the closest emergency area to land would probably change along the trajectory. This value could be automatically updated or not. It could even be evaluated and decided by the onboard computer considering the aircraft location and status, as suggested in [99].

- *Relative position*. The relative position of the aircraft in the current segment can be seen at a glance, under the remaining distance and time to the next waypoint. The next segment shows the minimum QoS on it if the current link is kept.

- *Flight plan*. At the right end of the strip, there is a button to open the flight definition, the sequence of previous pilots, notes introduced by these or any automated registry of events. When pressed, this information overlaps the CPDLC display. Strips are not covered with popup windows because at a given moment these could show emergency icons that should be always visible. As for the CPDLC messages, if received while the flight plan is checked, an icon appears in the strip and a sound is played.

- *Warnings and reminders*. Reminders and warning icons appear at the right end of the strip, sometimes accompanied by different aural indicators, depending on their urgency and type. When the message needs to be more attention-grabbing or descriptive, speech synthesis or notifications (described later) are used. Warnings associated with failures or unexpected readings should also be highlighted, especially for flights in the background.

  In the prototype, an example of reminder was implemented for the link update task designed for the exercises. Some time before reaching a waypoint where a link update is scheduled, a reminder is printed (Fig. 18, 1). This margin of time was ample enough considering that the planned new link could not be available or not with the expected quality and the participant would have to find an alternative. If the change of link is not completed after overflying the scheduled waypoint, a warning is printed to note that the aircraft is in the buffer area (Fig. 18, 2). If this area is exhausted, the link has not been updated and there is no coverage for the current frequencies (the change could haven been scheduled only to improve the QoS) the strip shows the lost link status (Fig. 18, 3) and in the prototype the commands and CPDLC panels become disabled. Later, after the established lost link timeout, the strip warns the pilot that the aircraft must be executing the procedure that was loaded when the link was lost (Fig. 18, 4). See them in action in [43], Screencast 9.



10:05:30 Logged as: pilot1

Callsign: BAG7430 Model: B733 Or.: EDDV Dest.: EINN Activation: 10:00:00
FL: 240 CFL: 240 RFL: 300 Heading: 266 IAS: 199.90 Mach: 0.60
Last Read: 10:05:30 Lost Link Timeout: 30 Freq.: 90.0 Lost Link proc.: A
OSN 0.4 km left to RKN SPY 103.8 km 137.7 km

Callsign: BAW962 Model: B733 Or.: EGLL Dest.: EDDV Activation: 10:00:00
FL: 340 CFL: 380 RFL: 370 Heading: 86 IAS: 216.82 Mach: 0.65
Last Read: 10:05:30 Lost Link Timeout: 30 Freq.: Lost Link proc.:
RKN 79.1 km left to OSN ROBEG 103.8 km 67.2 km

Callsign: CMM2433 Model: LJ35 //C650 Or.: EDDF Dest.: EGKK Activation: 10:00:00
FL: 330 CFL: 330 RFL: 400 Heading: 316 IAS: 226.80 Mach: 0.68
Last Read: 10:05:30 Lost Link Timeout: 30 Freq.: Lost Link proc.:
BIGGE 1.7 km left to HMM RKN 72.4 km 71.2 km

Callsign: DAT266 Model: MD83 Or.: EBBR Dest.: EKCH Activation: 10:00:00
FL: 310 CFL: 310 RFL: 330 Heading: 41 IAS: 226.01 Mach: 0.68
Last Read: 10:05:30 Lost Link Timeout: 30 Freq.: Lost Link proc.:
DOM 50.0 km left to OSN BASUM 72.8 km 71.7 km

Callsign: DLH3478 Model: A320 Or.: EGLL Dest.: EDDF Activation: 10:00:00
FL: 320 CFL: 320 RFL: 250 Heading: 160 IAS: 231.85 Mach: 0.69
Last Read: 10:05:30 Lost Link Timeout: 30 Freq.: Lost Link proc.:
DOM 54.6 km left to GMH FFM 63.5 km 134.9 km

Callsign: KLC223 Model: CRJ1 Or.: EHEH Dest.: EDDT Activation: 10:00:00
FL: 290 CFL: 290 RFL: 290 Heading: 86 IAS: 222.24 Mach: 0.66
Last Read: 10:05:30 Lost Link Timeout: 30 Freq.: 90.0 Lost Link proc.:
ROBEG 34.7 km left to DLE HLZ 41.8 km 63.2 km

Altitude: Heading: IAS: Proceed to:
ROCD: Turn: Mach:
C2 mode: Freq: Lost link plan:

KLC223 BAG7430 CMM2433
BAW962 DAT266 DLH3478

Attention! Close to low link area using current link.

REQUEST CLIMB TO 380
Sent: 10:00:28

AFFIRM
ALRT: M Received: 10:00:34

REQUEST
REPORT
CONFIRMATION
WHEN
FREE TEXT
ITP

28 **LEAVING** [level single]
29 **CLIMBING TO** [level single]
30 **DESCENDING TO** [level single]
31 **PASSING** [position]
37 **MAINTAINING** [level single]
41 **BACK ON ROUTE**

Code or Keyword:

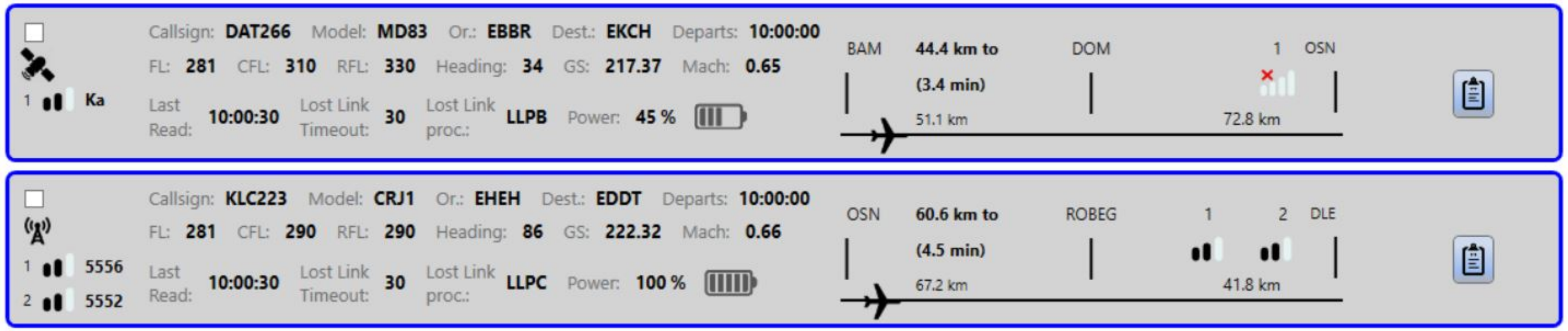
49 WHEN CAN WE EXPECT 220

ATC freq.: 0001

31 PASSING RKN

supervisor

Figure 16: Prototype NtoM Pilot Interface. 1) List of assigned flights; 2) Notifications and feedback requests; 3) set of commands embedded for experimental purposes; 4) CPDLC display; 5) NtoM network chatroom.



☐
1 Ka
Callsign: DAT266 Model: MD83 Or.: EBBR Dest.: EKCH Departs: 10:00:00
FL: 281 CFL: 310 RFL: 330 Heading: 34 GS: 217.37 Mach: 0.65
Last Read: 10:00:30 Lost Link Timeout: 30 Lost Link proc.: LLPB Power: 45 %

☐
1 5556
2 5552
Callsign: KLC223 Model: CRJ1 Or.: EHEH Dest.: EDDT Departs: 10:00:00
FL: 281 CFL: 290 RFL: 290 Heading: 86 GS: 222.32 Mach: 0.66
Last Read: 10:00:30 Lost Link Timeout: 30 Lost Link proc.: LLPC Power: 100 %

Figure 17: Strips of flights with satellite and terrestrial links showing the current link quality (left) and next segment quality if the current frequencies are maintained once past the next waypoint (right). The values represent the minimum quality along the segment or worst case.

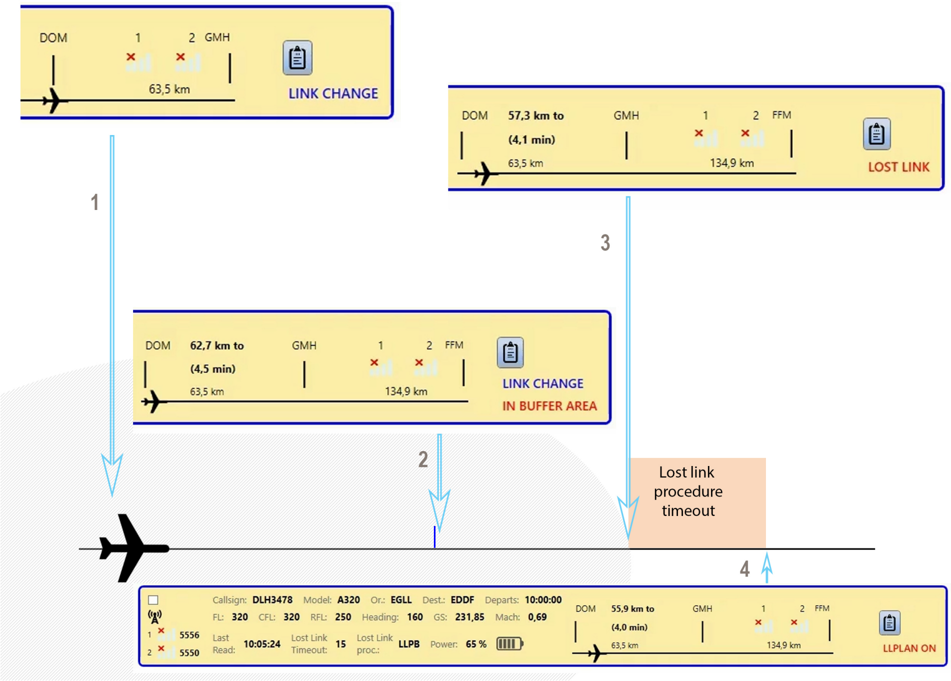
As it can be seen, strips also show a checkbox in their left end. This flight selector is used when the pilot wants to request a transfer of flights; they must enable the checkbox in each of the flights to be transferred and then press the Transfer Flights button at the top of the screen.

3.3.2 The CPDLC Display

In the prototype, the displays in the controller and pilot clients are almost identical; their difference appears only in the message element set, the hotkeys, the fact that the controller will not see a button to apply the instructions, and the margin for the timeout of the messages, which if quite greater for the controllers.

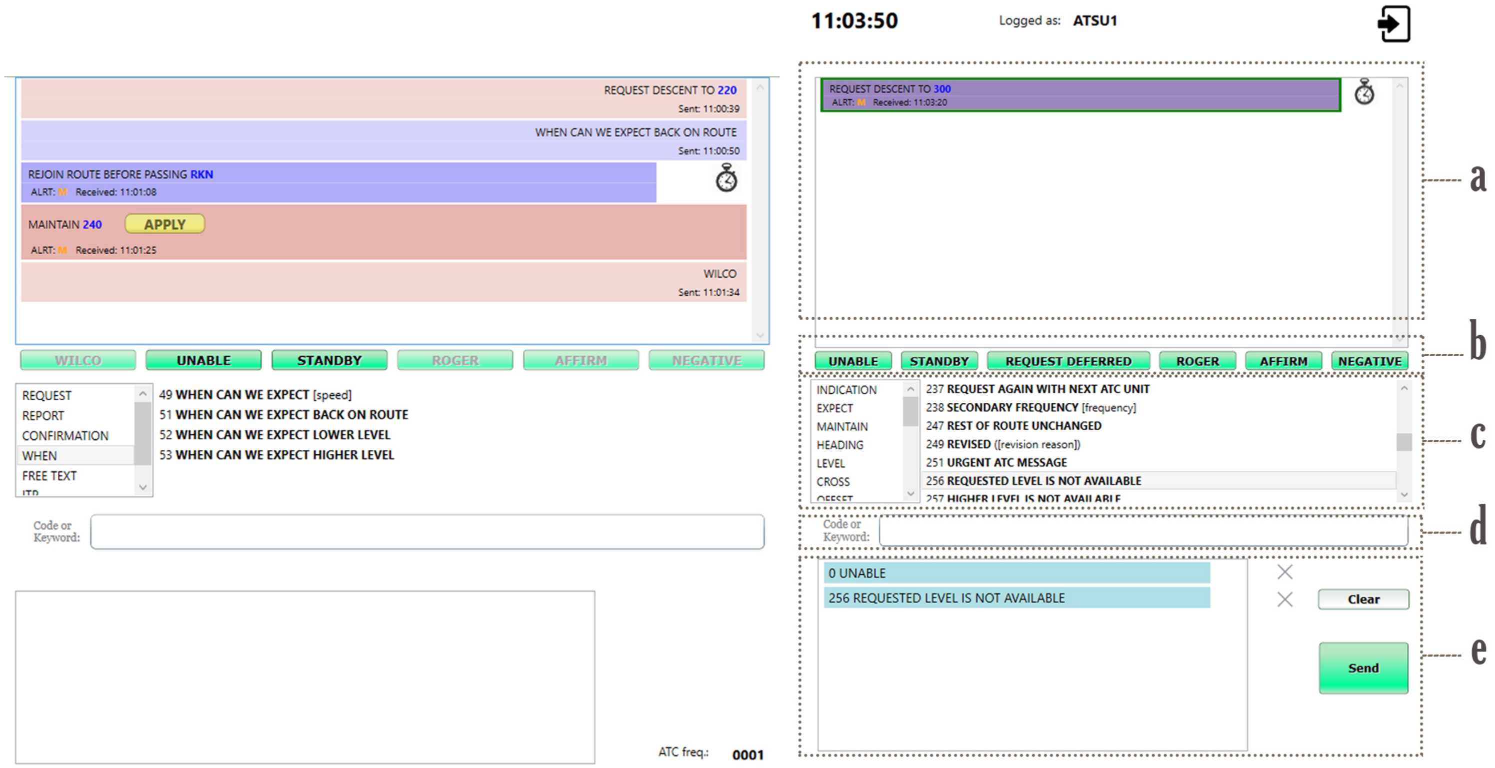
MESSAGE HISTORY To reproduce a familiar and therefore intuitive behaviour, the history stacks the messages mimicking a common chatroom application; the last one is shown at the bottom, automatically scrolling down when the list of messages overflows the window, meaning that the most recent message is always visible (Fig. 19, a). Having the log of messages within view helps to recall at a glance the conversation, with no need to navigate to load previous messages as in current displays. The horizontal alignment of the text also follows a typical approach: left for the received messages and right for the sent ones. Each dialogue is assigned with a random hue for the background of its messages to help distinguish them. Within each dialogue, messages sent are coloured with a lower saturation to highlight the received ones. See Fig. 1 or [43, Screencast 5] to check this message colour differentiation.

One benefit of this kind of message printing with respect to a usual cockpit CDU is that it eliminates a couple of identified problems. First, with multi-element messages, pilots can fail to read the whole message before responding to it, and those missed message elements could involve some condition to be met. An example of this problem and the impact in the tasks of the ATC is illustrated in the video [100]. Each of the stakeholders offers a solution to deal with this problem. Some display manufacturers disable the buttons to reply until the message is navigated and displayed to its end. Some airlines suggest to their pilots printing the long messages (a practice not recommended as printers are not always reliable [86]). And finally, some ATSUs restrict the messages to two maximum elements. In a representation like the one suggested here, these problems disappear, as the message is displayed in its entirety. In the unlikely case that the text could overflow the window (this could happen if the font size is increased a lot), the automated scroll to the bottom provides a clue about the extension of the message, that can then be navigated to the beginning using the scroll bar.



☐ Callsign: DLH3478 Model: A320 Or.: EGLL Dest.: EDDF Departs: 10:00:00
FL: 320 CFL: 320 RFL: 250 Heading: 160 GS: 231,85 Mach: 0,69
1 5556
2 5550
Last Read: 10:05:24 Lost Link Timeout: 15 Lost Link proc.: LLPB Power: 65 %

Figure 18: Reminders and warnings associated with the link update task.



REQUEST DESCENT TO 220
Sent: 11:00:39

WHEN CAN WE EXPECT BACK ON ROUTE
Sent: 11:00:50

REJOIN ROUTE BEFORE PASSING RKN
ALRT: M Received: 11:01:08

MAINTAIN 240
ALRT: M Received: 11:01:25

WILCO
Sent: 11:01:34

49 WHEN CAN WE EXPECT [speed]
51 WHEN CAN WE EXPECT BACK ON ROUTE
52 WHEN CAN WE EXPECT LOWER LEVEL
53 WHEN CAN WE EXPECT HIGHER LEVEL

ATC freq.: 0001

11:03:50
Logged as: ATSU1

REQUEST DESCENT TO 300
ALRT: M Received: 11:03:20

237 REQUEST AGAIN WITH NEXT ATC UNIT
238 SECONDARY FREQUENCY [frequency]
247 REST OF ROUTE UNCHANGED
249 REVISED ([revision reason])
251 URGENT ATC MESSAGE
256 REQUESTED LEVEL IS NOT AVAILABLE
257 HIGHER LEVEL IS NOT AVAILABLE

0 UNABLE
256 REQUESTED LEVEL IS NOT AVAILABLE

Figure 19: Pilot (left) and controller CPDLC displays (the situations depicted do not belong to the same communication). Parts: a) message history; b) hotkeys; c) message element selection by category; d) element selection by code or keyword; e) message composition.

In the messages, the variables of the element are printed in bright blue, to make the key part of the text stand out. Each message element is printed in a different row, and the whole message shows its timestamp, which is important to identify messages with a great delivery latency, showing instructions that are no longer valid or addressed to previous aircraft; the absence of this important missing data in the cockpit displays is pointed out in [101]. The alert level of the message is information of interest mainly for the controllers, who can use it to decide which message to attend to first, and is determined by the highest alert level among the message elements; it is printed in a different colour depending on the urgency. Accordingly, when a message is received, the kind of aural warning depends on its alert level.

Open messages are those waiting for an answer. These show an icon of a chronometer to indicate that the countdown to reply to that message is running (100 s for the pilots, 270 s for the ATC). If the timeout expires, the message should be ignored. Although initially considered, the option to print the remaining time to answer was discarded. A moving element attracts attention, and adding one on each open message was considered an increase of the displayed information and a source of stress. But, as learned from the first experiments, it seemed important to warn them that the time to reply was about to expire. Under high levels of workload, sometimes they would receive a message, postponed the reply to attend other flight, and then simply forget to send it. A first solution chosen was to play a synthetic voice warning that there were 30 s left for the timeout. Currently, some cockpit displays have an aural warning; in the case of the controllers, a visual blinking icon and a different colour for the flight tag in the radar. However, similar than in the case of the delay in the execution, those 30 s should not be a fixed value, and the tolerated delay and therefore the moment of the warning, could depend on the situation. Having the system access to information about the environment, and depending on the kind of instruction, different reminders could be associated to the open messages. A message in a situation that would require a quick reply or containing an IMMEDIATELY element, would be striking and insistent from the very moment it is received. A consensus should be reached with controllers about how to determine these differences. Nowadays, the use of CPDLC or voice communications depends on how quick is the reply required. The slow composition of the message relegates CPDLC to a secondary option, losing the benefits that it would provide as first communication method. Having a quicker display, like the one suggested here, CPDLC could be used in more situations, and then it makes more sense to highlight when the reply should be immediate or if there is more flexibility on the time to answer.

QUICK REPLY HOTKEYS Buttons for some common or straightforward replies have been placed under the history to allow a quick reply: WILCO, UNABLE, STANDBY, ROGER, AFFIRM and NEGATIVE for the pilot; UNABLE, STANDBY, REQUEST DEFERRED, ROGER, AFFIRM and NEGATIVE for the controller (Fig. 19, b). These hotkeys appear active only when they make sense considering the selected message to be answered. In Fig. 20 the pilot receives a message that, considering the reply precedence of the elements, should be only replied with a WILCO/UNABLE (W/U) or a STANDBY. It should be noted that the hotkey labels match the final message element that the controller will receive, instead of the practice found in some CDUs in which a key is associated with several meanings, like ACCEPT as a positive reply representing the replies WILCO, ROGER and AFFIRMATIVE. The use of dedicated buttons is more aligned with the semantics of the reply, reducing ambiguity. The message elements present in these hotkeys are also available like any other element in the search by category or keyword filter, as they could be combined in multi-element messages. It would not be difficult to allow the users the option to add, remove or change hotkeys with frequently-used instructions, even when they contain parameter values, while there is space available (a controller considered this already populated messages as an interesting option as long as a confirmation dialogue was showed before applying them).

MESSAGE ELEMENT SELECTION BY TYPE The statement in [36, Chapter 3] — *"The pilot interface to the data link system must be efficient and easy to operate. Pilot-controller messages require some rapid entry mechanism"* — was taken seriously into consideration for the implementation of the process of composition. The first step when a user wants to write a message will depend on whether it is the first message of a dialogue or is an answer. In CPDLC, messages are linked forming dialogues. Open messages are those waiting for a reply, and an open dialogue is that with an open message. If the first message of a dialogue requires a reply, that reply will close the message, but the dialogue will only be closed if that reply does not require a reply too. It is possible for more than one dialogue to be open at once, so to answer an open message, it is necessary to indicate which one is to be dealt with first. This is done by clicking on the message in the history, which will make it appear framed (Fig. 20). The next step is to choose a message element. A message can contain up to seven message elements. The display offers two ways to select them. The first is to use the menu list of categories appearing at the left side of this block (Fig. 19, c). These categories are determined based on the description of the elements provided in the documentation, although users could be surveyed about other categorisations preferred or the interface could allow their own customization. When the user selects one of the categories, the list of elements it contains appears on the right. These are represented by their syntax, with the parameters in square brackets and the optional parts of the element in parenthesis. This kind of element selection is useful when the user is unsure about the elements available. If a pilot wants to send a report but cannot remember the code number or any keyword on it, selecting the Report category shows the list of this kind of elements available. One idea not implemented in the prototype is to somehow provide a short description of a selected element. A clear definition of the meaning of an element and its possible incompatibilities would be handy and could help prevent errors.

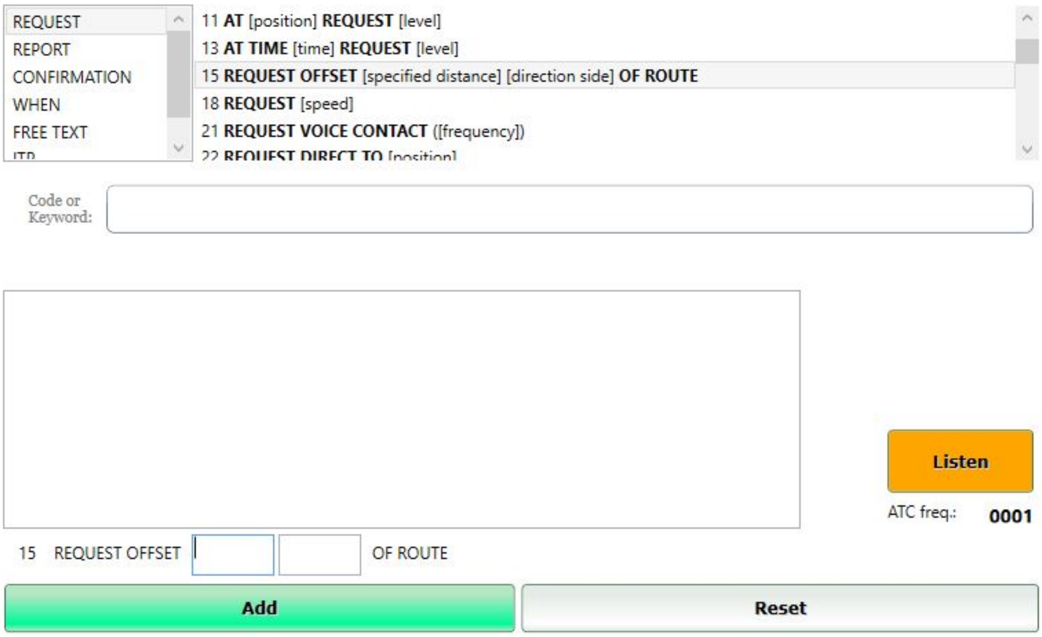
MESSAGE ELEMENT SELECTION BY FILTER The second selection option is the input box labelled Code or Keyword (Fig. 19, d). On it, the user can begin to write the element code or any of the words that he or she remembers appearing in it to receive a list of filtered elements containing those characters. Check the selection modes in action in [43, Screencast 6].

CROSS DOM AT 330
REPORT MAINTAINING 330
ALRT: M Received: 10:03:00

Figure 20: Only those hotkeys that would be valid replies appear enabled when the message to be answered is selected.

MESSAGE COMPOSITION Once the element has been selected, if it has parameters, a form appears at the bottom of the display, with an input box per parameter (Fig. 21). At the moment, these inputs allow any text because there was no expert advice available during implementation on the specific values, step or ranges for these variables. When possible, limiting input options with a dropdown would be desirable, as it reduces the possibility of errors, the head-down, and the need to move the hand from the mouse to the keyboard to write values. Some values allow different units; for instance, the level can be specified *"as a single or block level in feet, meters or flight levels"* [102], and a text box forces the pilot to write the unit abbreviation, which is slower than selecting it from a list and error-prone. Regarding numerical values, a selector makes sense when the range and step do not result in excessive scrolling to find the value. When the values have been filled in, the Add button includes the element in the message. If the element does not contain any parameter, it will be directly included when selected. The user can drag and drop the elements in the message composition box to change their order, and delete them using the cross located to the right of each one (Fig. 19, e).

GUIDANCE AND ERROR CHECK In the same way that it cannot be expected that users will be able to remember the syntax of the large message element set, with around 150 elements for downlinks and 330 for uplinks, it must be assumed that it will be difficult for them to memorise all combination incompatibilities or non-recommended practices. In addition, the input value of variables must be checked (like numbers when letters are expected, unspecified units, or values out of range) when the values cannot be chosen from a list. These kinds of warnings will be triggered when the user tries to add an element or send the whole message. Some warnings will avoid the sending of the message, while some others could just inform of a detected ambiguity, or a practice that is not recommended, suggesting a better alternative. Considering the huge task represented by implementing all these warnings, which would require also advice and expert review, just some illustrative examples were included in the current prototype. Some can be seen in [43], Screencast 6.



11 **AT** [position] **REQUEST** [level]
13 **AT TIME** [time] **REQUEST** [level]
15 **REQUEST OFFSET** [specified distance] [direction side] **OF ROUTE**
18 **REQUEST** [speed]
21 **REQUEST VOICE CONTACT** ([frequency])

ATC freq.: **0001**

15 REQUEST OFFSET OF ROUTE

Figure 21: Each element variable has its own input box, the value of which is subject to its associated error checks.

ADAPTIVE MESSAGE ELEMENT SET AND RULES Another interesting feature would be to automatically update the display behaviour and available message elements set depending on the service provided by the ATSU. The most basic case would be when the aircraft sends the Data Link Initiation Capability (DLIC) initiation message (a kind of request for service and login) and the DLIC ground system provides the aircraft with information about the data link applications available (Context Management, CPDLC, Automatic Dependent Surveillance-Contract). If the ATSU does not provide CPDLC services, the display would not allow them to be requested, warning the pilot that the service is not provided. If provided, it must be considered that just a small subset of message elements would be available during the gradual implementation of the service, and it can change among ATSUs of even the same CPDLC version, as there are mandatory and optional services to be supported. Then, the set can be partially determined by the CPDLC version to be used, with this data being provided by the ATC centre during the aircraft login. In the U.S., the service for the domestic en route environment started in 2019, and its message elements include: altitude and re-route clearances, crossing restrictions, pilot requests and emergency messages; there are differences between oceanic and domestic use, and free text messages are not allowed (a detailed list of the elements allowed appears in [86]). By contrast, Europe uses a message element set defined in the Link2000+ Programme [103], which is a subset of the ICAO's CPDLC [29]; on it, some elements are mandatory to support by the European air traffic service (ATS) authorities providing CPDLC and others are optional. For instance, the Canarias FIR [104] uses it for any kind of instructions: climb, descend, change of frequency, et al; free text messages are allowed (but not recommended). In continental area, the Maastricht Upper Area Control Centre handles a wider message element set (detailed in [105]) than the one envisioned in the U.S., but free text messages are not allowed either. While an unsupported request would be rejected with an error message (UM162 MESSAGE NOT SUPPORTED BY THIS ATC UNIT), it seems convenient to implement a proactive display aware of the limitations of the area, considering that it is information that is publicly available, to avoid useless efforts. Consequently, the display would hide those elements that are not allowed or would limit the input units, which are also affected by the version. The appropriate behaviour would be updated with each change of CDA. In some cases, it would just warn about unsuitable use. For instance, CPDLC is not recommended below 10,000 feet but is allowed to back up any possible problem with voice communications.

A similar approach can be found in [106] for a cockpit display feature called the CPDLC version abstraction functionality; it suggests seamlessly filtering the message elements displayed depending on the application version received during the establishment of the session. However, the adaptability suggested for the NtoM display, pretends to be more fine-grained, taking into account the ATSU-specific optional elements and warnings, which would be possible by keeping a database of CPDLC service availability at an ATSU level instead of at a version level. This would result in a precise fit to the supported services.

RESPONSE TIME MITIGATION As mentioned, one problem that could arise during multi-RPAS operations is an added increase of the response time to ATC if the pilot is busy with some other aircraft. CPDLC messages have a reply timeout associated and, when exhausted, it invalidates the message received. When this happens, the controller could try to send the message again or revert to voice. Either way, the result of missing a reply means overworking controllers. Following are the measures to try to keep fluid communications and avoid these situations. These features are conceived as part of the NtoM framework, requiring access to the aircraft telemetry and other data, so, in principle, would not be part of a standalone implementation of the display.

- *Automated reports*
  Apart from the information provided by ADS-B, the Automatic Dependent Surveillance – Contract (ADS-C) data link application allows the ATC to request or schedule customised reports from the aircraft. The whole process is managed by the aircraft system and the pilot is not involved or even aware of it. But, aside from these, ATCs can ask for CPDLC reports. To reply to these, the pilot should look for the data required wherever it is displayed, compose the message and send it when required, as the moment of the reply could have some event associated, like the case of a UM130R REPORT PASSING [position]. These reports require sending a value that can be retrieved or calculated having access to some sources or readings, like the global positioning system (GPS) receiver or the inertial navigation system (INS). This is the case with elements UM35, UM36 and UM37 (CONFIRM ASSIGNED [LEVEL/SPEED/ROUTE]), or REPORT ENDURANCE AND PERSONS ON BOARD. This kind of report could be automatically replied to in order to release the pilot from the task. The ones requiring a condition to be met before being sent, like UM130R, need monitoring to determine the moment. The screencast 3 in [43] shows an example of both of these kinds of reports: the

automated reply of a Cleared Flight Level report request, and the conditional automated report when it reaches a certain level.

A proposal with common points with this feature can be found in [107]. It considers that a flight plan could be designed associating the sending of CPDLC messages to waypoints, and suggests a display representing the route with tags associated with the waypoints with those scheduled messages. Those communication requirements could have been added during the flight by the pilot or as a consequence of CPDLC messages received from the ATC. When the moment to send the message arrives, the proposed subsystem asks the pilot for confirmation to send the message if it is a report or warns about a message pending to be composed. This message would appear in a dedicated display or in the MCDU. The confirmation dialogue would not require filling in any data, as the unit would have access to the sources of information required. The NtoM system uses a more general concept of tasks (e.g. a scheduled frequency change) that includes the communications as any other planned task and could ask for confirmation or not depending on the type of action.

- *Loadable messages*
  Some messages containing input data for the FMS could be directly loaded on it, improving efficiency and safety by avoiding input errors from the flight crew ([36] Part I, 3.6). It also allows to reduce the workload and provide a quick execution. This loadable feature is something that can already be found in current data link cockpit displays. Then, when possible, an Apply button can be found next to the received message in the pilot display. This would send to the aircraft with a single click the commands required to execute the orders contained in the message.

  The possibility to add the button to a message, and the commands that result from its interpretation must follow some rules. For instance, as specified in [102], when an uplink consists of multiple message elements, they must be loaded in the FMS following the order in which they were composed, even when this means that an element overwrites the behaviour of a previous element. A consequence of this is that only one Apply button will be placed for the whole message. With a button for each element, the pilot could load an unexpected order of commands if pressed in a different sequence. In any case, controllers are encouraged to not send more than one independent or unrelated clearance per message to allow the flight crew to respond to them individually [29, Chapter 4.3.6] and avoid any ambiguity. If a single uplink message contains more than one clearance and the crew cannot comply with any of them, it forces them to reply UNABLE to the whole message; and the opposite also applies, with a multi-element downlink with several requests that cannot all be cleared. However, more than one clearance per message is allowed to reflect the case of the dependent clearances, where the instructions must be followed only if all requirements in the message are met.

  In those messages allowing the Apply button and that do not require a further reply, pressing the button assumes an implicit WILCO, that will be automatically sent; for instance, in the case of a message with a unique element UM20 CLIMB TO REACH [level]. However, the pilot could receive a multi-element message like:

UM74 PROCEED DIRECT TO [POSITION]
UM148 WHEN CAN YOU ACCEPT [LEVEL SINGLE]

On it, the highest response precedence is a W/U, but the pilot still needs to send an answer for the UM148 element. Here, the guidelines allow two options:

a) Sending a multi-element message where the first element would be the response to the element of highest precedence (UM74, then a W/U), and a second element that would respond to the UM148 (a DM83 WE CAN ACCEPT [speed] AT TIME [time], a DM84 WE CANNOT ACCEPT [speed], or a DM116 WE CAN ACCEPT [speed] AT [positionR]).

b) Sending a first message with a single element for the element with the highest precedence and later sending a separate message with the answer to the other element.

Implementing b) would allow an Apply button be put in place to execute the direct to when pressed and then automatically send the WILCO; then the system can just hope that the pilot will not forget to compose and send the second message. This hope comes from the fact that sending the WILCO closes the whole message, and the dialogue, even when the controller is waiting for the reply to the UM148. So, the open message icon and the voice reminder would not be there to claim for the second message. To solve this matter, a different kind of warning could be implemented, a kind of pending message alert for closed messages. But increasing the number of different warnings would increase the display complexity, and allowing those separate messages seems error-prone. In any case, if the reason why a pilot could prefer to send separate messages is the need for some more time to reply, the STANDBY message can be used for that purpose. Then, the decision was to force the first behaviour; the system will issue a warning if any of the elements does not have its required reply. This is illustrated in [43, Screencast 4].

- *Voice-controlled composition and commands*
  One way to release the requirements of visual resources in an interface is to move some of the information to the aural sense, although this is frequently done to underline some displayed event. This alternative and backup of auditory information is useful when the user is busy attending to a task and cannot move the sight to another source of information, particularly when the access to the information is slow. An illustrative example is the proposal [108] of a module to read aloud just-issued pilot reports (PIREP), weather and separation information, or even CPDLC messages, with the option to apply filters to these messages.

  A message composition alternative partially developed in the prototype uses voice recognition. The pilot could be busy with a flight, with the need to keep looking at the GCS when a message arrives to other aircraft. If he/she decides to wait a bit to read the message, it would add extra stress to their current operation and a delay in the response to the ATC. The interface offers the possibility to avoid that delay by verbally composing and sending the reply. When the message arrives for the flight in the background (a flight that is not the one currently selected, so the message cannot be seen in the CPDLC display), a synthesised voice reads it aloud, identifying the aircraft receiver by its callsign. If the pilot does not want to verbally compose a reply right then, maybe because needs to check some data previously, an "ok" will dismiss the voice assistant. At that moment, at least, they already know the kind of message and how urgent it is, and the decision about the reply can be advanced during the flight swap process. If the pilot wants to reply, just says the elements in the desired order and then says "send". No ending keyword is required at the end of each element; the grammars describing the syntax of the elements allow the engine to determine their pattern. For an element like UM238 SECONDARY FREQUENCY [frequency], the engine will consider that the element is complete after it hears a frequency number. The grammars also reflect the optional parts of the elements. The assistant will read back the message for the pilot to confirm that it was correctly recognised and, after the confirmation, will send the message. This kind of composition is especially useful for simpler replies that do not require a previous evaluation of the status of the aircraft or the flight plan, which would require a flight swap anyway. A basic case is shown in [43, screencast 7]. The voice recognition implementation for the pilot interface, including the dictation of the CPDLC messages, was well advanced, but the remaining grammars associated to the message parameters were too time-consuming and there was no advice available to clear up many doubts about the phraseology of the units and the suitable ranges. Besides, the performance of the speech recognition engine used (Microsoft's) was not satisfactory enough and other options should be tested. For these reasons, this feature was discontinued and left as future job. The voice control has been suggested as a safety measure in previous research, also to be used by controllers [109] [110].

An operative implementation would require the involvement of final users to find a balance between the phraseology needs and the limitations of voice recognition. A verbal composition that would require too many steps and clarifications for an unambiguous input would not provide any advantage with respect to other solutions in the graphic display. Some issues should be agreed, like how to specify the units of the parameters, how to unequivocally pronounce names or all those involving voice recognition and detailed in [111], which clearly illustrates this challenge when used to allow ATC to compose CPDLC messages. As mentioned in that work, from the controller's side, used to compose at least the simplest messages could be especially interesting in those scenarios where a mix of voice and data link communications would force them to constantly swap voice and keyboard inputs, which adds an undesired increase of physical demands and perceived workload, as described in [112]. That situation could be suffered, on a smaller scale, by the multi-RPAS pilot with several aircraft overflying areas both with and without

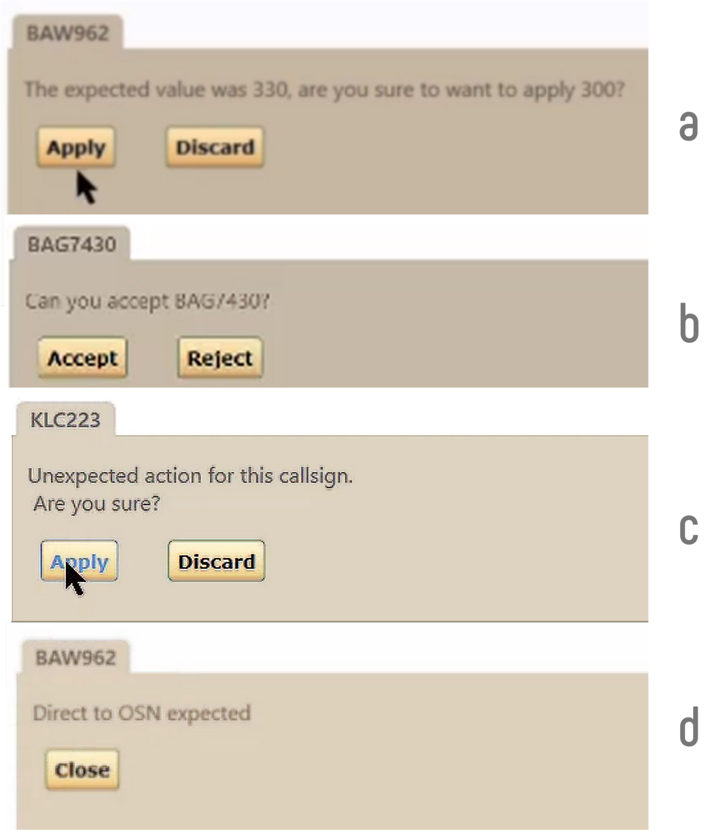
CPDLC support. [113] also advocated for ATC CPDLC verbal composition as a tool to reduce communication errors, and [114] describes a detailed experimental analysis of the real use of ICAO recommended phraseology to shed some light on the conditions that any feasible speech recognition technique must deal with.

3.3.3 Notifications

Notifications (Fig. 16, 3) are reserved for warnings or reminders requiring a detailed description or those situations where a dialogue is required to get feedback from the pilot. The first case is that where the system requires printing a detail of actions or values. The latter is the case of the offers for flight assignments, requests for feedback (deadman pedal), confirmation of unexpected actions, etc. Fig. 22 shows some of these examples.

3.3.4 Control Migration

A pilot selected to be assigned with a flight receives a dialogue to accept or reject it. Pilots could set their status to not available to receive these offers, for instance, while attending an urgent situation that the system could not detect during the evaluation of the best candidate. Other situations could set this status, like the system detecting a pilot indisposition or GCS malfunction. If accepted, a new strip for the received flight appears, framed in a different colour to highlight that it still cannot be controlled, but the readings are updated, and the pilot can access the flight plan and the history of CPDLC. When receiving a new flight, the pilot needs to study the flight plan and create a mental picture of the flight status, pending and next tasks, and fit this new requirement of attention along with the rest of the current flights, if any. A plus of stress is added if the flight has been transferred during an emergency, like pilot indisposition, and requires rapid actions. This applies also to the one-to-one pilot-aircraft ratio. There are some things, though, that can be done to alleviate the risks of these transfers. First, the receiving pilot must be provided with a clear summary of the situation of the aircraft, a history of the communication exchange and a highlight of pending actions or abnormal situations. Instead of relying on voice communications (transferring pilot could not be available), such record of events would be provided by the interface. To this respect, the NtoM CPDLC display provides a quick and clear log of the ATC communications, with the highlight of the open messages ([29] 5.1.2.4: [...] *"the flight crew carrying out the 'handover' briefing should thoroughly brief the 'changeover' flight crew or flight crew member on the status of ADS-C and CPDLC connections and messages, including a review of any pertinent uplink and downlink CPDLC messages (e.g. conditional clearances)."*). In the NPI, the summary of events, the flight plan and any annotations provided by previous pilots, are envisioned to appear in the area ocuppied by the CPDLC display, when the button with the clipboard icon in a strip is pressed. In the current version, it only prints the flight plan and the list of previous pilots, that way the receiving pilot can check the transferring pilot and use the chatroom to coordinate the transfer or clarify any doubt. While



The expected value was 330, are you sure to want to apply 300?

Can you accept BAG7430?

Unexpected action for this callsign.
Are you sure?

Direct to OSN expected

Figure 22: Some types of notifications.
a) "The expected value was 330, are you sure you want to apply 300?". the pilot introduces a value for the command that does not match that indicated in the flight plan or in the instruction of the controller and the interface asks for confirmation.
b) "Can you accept BAG7430?". The pilot is offered an assignment of flight.
c) "Unexpected action for this callsign. Are you sure?". The pilot sends a command that is not scheduled in the flight plan nor has been requested by the controller, probably selected the wrong flight, so is asked for confirmation.
d) "Direct to OSN expected". First warning after exhausting a timeout for a tolerable delay in the execution.

The a) and c) only apply in the configuration of the prototype, where the commands are sent to the aircraft from the server and not directly from the GCS (i.e. we can stop and ask for confirmation before execution).

not implemented in the prototype, the interface could allow pilots to introduce annotations available to the following pilots. The receiver pilot, once prepared to control the flight, informs the transferring pilot, who executes the transfer. The receiver can then control the new flight and the previous pilot sees the flight strip disappear from the NPI. Screencast 10 in [43] illustrates the process.

Finally, if considered useful by controllers, and especially when the handover is the result of an emergency, the system could alert them of the transfer process for a possible delay on the communications or instructions execution. It could even broadcast a warning to the traffic nearby. These measures would be automatically triggered during the transfer, with no need to overload the pilots.

3.3.5 Intra-NtoM Communications

The chat panel allows users to communicate with any pilot and supervisor connected to the NtoM network (Fig. 16, 5) (the use of instant messages for coordination is already common in military operations [115]). It can be used by pilots to provide information or ask for advice from their colleagues. Pilots involved in a flight transfer can coordinate the handover through it. Supervisors can ask the pilots for feedback about actions, or to check the availability to accept an assignment. While *live* verbal communication — like phone calls — could be added to the interface, one of the benefits of the text messages is that is quite less disruptive and suitable for a multi-RPAS scenario. If the pilot is busy, the message can be attended later; meanwhile, the sender appears highlighted in a different colour, to note that it has an unread message associated. If the pilots consider that writing text messages imply a too high head-down, especially for long communications, a solution could be the possibility to record and send voice notes. A minus against voice communications is that the pilot cannot quickly come back and check a data provided on it, they would need to note it down during a live call, or play again the voice note. A workaround would be to rely on voice recognition to compose the message or add the possibility to get a transcription of the voice communications.

4 EXPERIMENTS

To affirm that a pilot controlling highly automated aircraft, with enhanced communications, workload balance, task assistance and monitors is able to safely operate several RPAs, their performance during such operations should be checked in a faithful simulation. These tests would require RPA pilots in missions similar in duration and complexity to those to be carried in real life. The GCS used should provide an API for the NPI to send and receive information and commands. The workload profile of each pilot should be available too. Controllers or any other role that could participate, like payload operators, should also be added with a realistic environment. Having all those resources available was far from possible, but a prototype running abstractions of a real usage could still reveal some weaknesses and suggest improvements that would also arise in a faithful recreation, especially with respect to the usability of the NPI or the management of the workload.

In any case, even having access to the lacked logistics and resources, a long-term concept like this is considering situations that do not exist yet, like regular commercial unmanned service integrated into the civil airspace. It is being assumed that there will not be special treatment for the RPAs in the ATM, but considering the differences in their capabilities with respect to manned aircraft, maybe finally they are subject to specific conditions if their operations could affect the performance of the manned traffic. Besides, the assumed full use of CPDLC as the main way of communication is still far in time, and probably even real pilots and controllers participating in the experiments should be trained on it first, which makes no great difference with respect to using non-professional pseudo-pilots to test the prototype.

The results from the experiments described here helped to build the prototype, inspire new measures and improve weaknesses. They provided insights about the mental mechanisms to handle the workload and how to improve the usability to manage the multiple flight control. To get consistent values, participants had to be present in some previous sessions to learn the procedures, the use of CPDLC and practice the use of the interface. Then they would perform some exercices with and without some of the aids implemented, because the experiments tried to reflect the impact in the performance while operating with or without the safety and usability measures. All that meant several hours of work along months, and only four people participated in all the tests.

4.1 FIRST SET

To build some of the features of the concept, it was necessary to build a representation of the workload associated with the tasks and their values (workload level, time). That way the prototype could show how the system keeps a forecast of workload and how it is dynamically updated due to ATC communications, changes of speed or trajectory ([43], Screencast 1). There was also the need to determine the top threshold of workload for a pilot and detect incompatibilities among tasks. These values served as a reference to design demanding test scenarios which helped to reveal weaknesses and inspire improvements that were addressed in the following stage of development. The overloaded multi-RPAS scenarios of this set of experiments allowed to detect monitoring needs and inspire improvements in the workload management and the usability of the pilot interface. Some of these findings were later implemented and put to test in the second set of experiments.

Design and Execution

The NtoM prototype was adapted to serve as a synthetic task environment to evaluate the requirements of time and attention of pseudo-pilots based on their performance while executing the tasks and task overlaps. To avoid having a person in the role of the controller, the ATC requests were automated, following a script determining the moment, message element and parameters of the messages that would be sent to the pilots. This allowed the identical simulation for every pilot of specific demanding situations, like receiving several requests at once or very close to each other, and checking how pilots deal with it. Considering that the participants would not have knowledge about piloting, the GCS that in a real implementation would be plugged to the NPI, was reduced to a minimalist panel of commands and embedded in the client. The flight plans handed to the participants were printed intuitive diagrams depicting the tasks they had to perform when reaching some of the waypoints (Fig. 23, top). These showed also the links and lost link procedures (LLP) projected during the planning of the flight. Participants were encouraged to mark the tasks as completed when done and write down any change in planned frequencies or LLP during the flight; not only to better organise themselves but also to communicate the changes during the control migration exercise.

The tasks to be executed by the participants did not try to reflect real-life procedures, they were artificially designed to represent different sets of actions and cognitive resources. However, they tried to represent the usual tasks required, as advised by a subject matter expert and military RPAS pilot. While some more were practised by the participants (requests for change of altitude, speed, handover), the ones finally included in the measurements of this first set of experiments were the following:

- *Change of Command, Control and Communications (C3) link*
  A task scheduled in the flight plan associated with the overflight of a waypoint. Fig. 23 top shows three tasks of this type. Before applying the scheduled change, participants must check that the strength of the new link is as expected. They do not have access to a map depicting the coverage of each frequency but the interface shows the minimum quality between the past and next waypoint, and between the next and following waypoint (Fig. 17). For simplicity, the quality of the link was summarised as a value from 0 (no

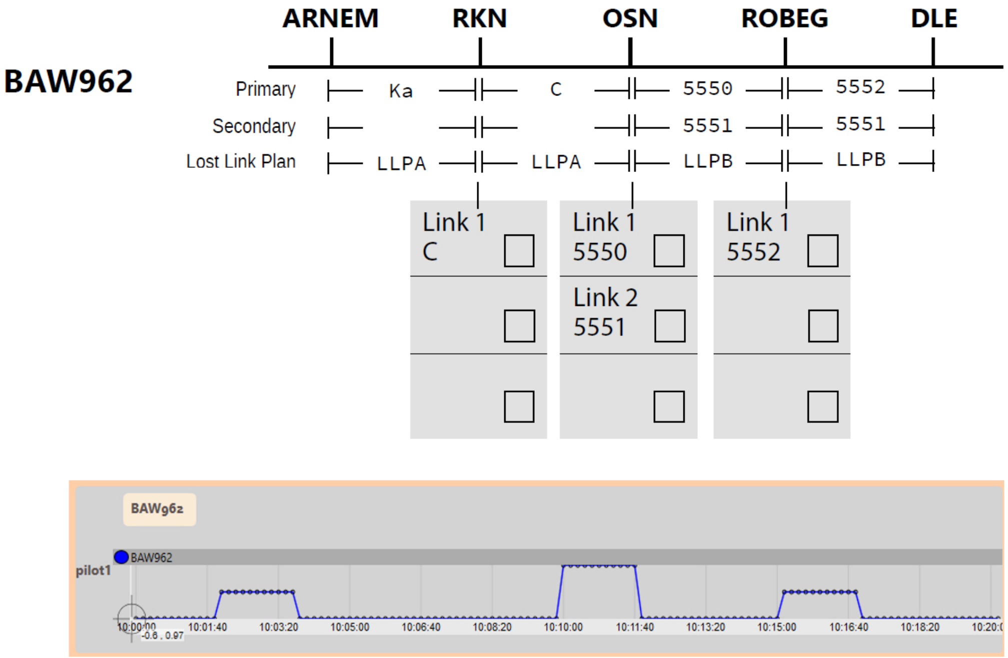


Figure 23: Top: printed diagram of scheduled tasks for a flight handed to the participants. Bottom: workload timeline for the previous tasks as seen in the supervisor client.

coverage) to 3. During some of the link updates, they would find that the quality of the planned link was 0, which meant that the frequency was unexpectedly unavailable and therefore they had to evaluate another frequency or band.

Ideally, in the prototype, if the diagram indicates changing the link when reaching a waypoint, this should be done just after passing it. If they do it too late, the aircraft could exit the buffer area. This is an area where the previous and next frequencies coverages overlap, providing a margin of time to change the link. If exhausted, they could lose the connection. This just a possibility because a change of link does not necessarily mean that the previous frequency is out of coverage in the next segment, the change could be just to improve the signal quality. Also, the change should not be too soon to not exceed the LLT value when there is no coverage for the new link in the segment left behind. The participants would learn with the practice that the evaluation of alternative links when the scheduled frequencies fail, requires time, so they would begin to check its availability a while before reaching the waypoint, to avoid exhausting the buffer area. This is why we find a pre and post time value in the time window for this task. The other two tasks are triggered by events, so they only have a post time associated.

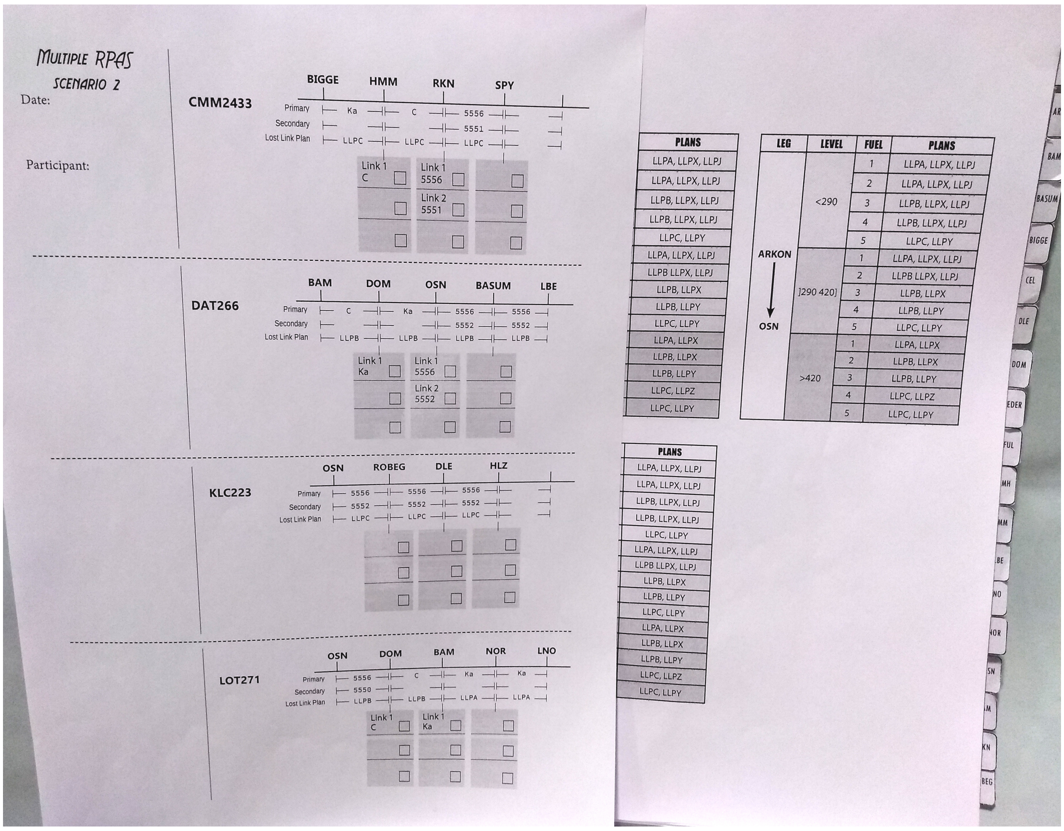
For the link, they could select directional antennas or satellite. With the first option, they had seven frequencies and had to try to always keep the primary and secondary links alive and with the maximum quality possible, especially the primary link. The ground link was only lost if the primary and secondary links had 0 quality. By satellite, they had two bands to choose, without a secondary link. Before applying the change of link, after choosing an option in the dropdown, a preview shows the minimum quality of the link along the current and next segment if the selection is applied. Fig. 17 represents the link qualities in a similar way to the preview (although the preview is shown as text); the out of coverage icon for Ka between DOM and OSN means that, while it could have a decent quality in the rest of it, at some point in that segment the quality is 0 and hence Ka should not be used in that segment. Between BAM and DOM its minimum quality along the segment is 2.

- *ATC instruction to change the current LLP*
  In the prototype, the LLP that would be loaded in the aircraft FMS, is automatically updated during the flight according to the flight plan or assuming a dynamic contingency evaluation [99]. But in the experiments, the controllers eventually asked the pilots to change it (f.i. suppose the alternative airport it suddenly not available) to create a demanding situation. Participants were handed a printed chart (Fig. 24), where they had to find and select an alternative plan depending on the current position, altitude, and fuel/battery level [20]. They were guaranteed that this alternative plan would always exist, so they could always reply WILCO to these instructions. After the change of the LLP, they had to send a free text message to the controller with the new plan selected, simply identified by its name in the exercise.

- *Direct to clearance*
  This contrived task was conceived as an example of especially demanding task. Assuming that previously the pilot asked for it, the participant receives this kind of clearance. Before sending the answer, the pilot must check if there is a link available for the segment between the current position and the destination waypoint (in a real situation, we can expect that this would be done before requesting the clearance). If there is a link available, after sending a WILCO and applying the command to lead the aircraft to the destination waypoint, the participant must check if there is a need to change the current LLP to one specific for the shortcut. If it needs to be changed, as in the previous task, the ATC must be notified with a free text message.

The first sessions were addressed to determine the values of time and attention required for the previous tasks. Lacking an objective measurement of the cognitive workload, like biometrics, it was substituted by subjective values based on the observation of the participants' performance, considering the number and kind of actions required and the dedication or attention observed. In a real implementation, contrary to these experiments, these values would be personalised for each pilot, but due to the few opportunities to measure each task for each pilot, it was considered more suitable to calculate an average requirement cost from the performance of all the pilots.

For simplification, and contrary to what would be more interesting in a real implementation, each task was taken as an indivisible unit, even if it could be divided in subtasks. The time amplitude of a task was measured as an uninterrupted period between the first and last interactions with the interface to execute



MULTIPLE RPAS
SCENARIO 2
Date:
Participant:

CMM2433	BIGGE	HMM	RKN	SPY
Primary	Ka	C	5556	
Secondary			5551	
Lost Link Plan	LLPC	LLPC	LLPC	

Link 1 C ☐ | Link 1 5556 ☐ | ☐
☐ | Link 2 5551 ☐ | ☐
☐ | ☐ | ☐

DAT266	BAM	DOM	OSN	BASUM	LBE
Primary	C	Ka	5556	5556	
Secondary			5552	5552	
Lost Link Plan	LLPB	LLPB	LLPB	LLPB	

Link 1 Ka ☐ | Link 1 5556 ☐ | ☐
☐ | Link 2 5552 ☐ | ☐
☐ | ☐ | ☐

KLC223	OSN	ROBEG	DLE	HLZ
Primary	5556	5556	5556	
Secondary	5552	5552	5552	
Lost Link Plan	LLPC	LLPC	LLPC	

☐ | ☐ | ☐
☐ | ☐ | ☐
☐ | ☐ | ☐

LOT271	OSN	DOM	BAM	NOR	LNO
Primary	5556	C	Ka	Ka	
Secondary	5550				
Lost Link Plan	LLPB	LLPB	LLPA	LLPA	

Link 1 C ☐ | Link 1 Ka ☐ | ☐
☐ | ☐ | ☐
☐ | ☐ | ☐

PLANS
LLPA, LLPX, LLPJ
LLPA, LLPX, LLPJ
LLPB, LLPX, LLPJ
LLPB, LLPX, LLPJ
LLPC, LLPY
LLPA, LLPX, LLPJ
LLPB LLPX, LLPJ
LLPB, LLPX
LLPB, LLPY
LLPC, LLPY
LLPA, LLPX
LLPB, LLPX
LLPB, LLPY
LLPC, LLPZ
LLPC, LLPY

LEG	LEVEL	FUEL	PLANS
ARKON ↓ OSN	<290	1	LLPA, LLPX, LLPJ
		2	LLPA, LLPX, LLPJ
		3	LLPB, LLPX, LLPJ
		4	LLPB, LLPX, LLPJ
		5	LLPC, LLPY
	]290 420]	1	LLPA, LLPX, LLPJ
		2	LLPB LLPX, LLPJ
		3	LLPB, LLPX
		4	LLPB, LLPY
		5	LLPC, LLPY
	>420	1	LLPA, LLPX
		2	LLPB, LLPX
		3	LLPB, LLPY
		4	LLPC, LLPZ
		5	LLPC, LLPY

PLANS
LLPA, LLPX, LLPJ
LLPA, LLPX, LLPJ
LLPB, LLPX, LLPJ
LLPB, LLPX, LLPJ
LLPC, LLPY
LLPA, LLPX, LLPJ
LLPB LLPX, LLPJ
LLPB, LLPX
LLPB, LLPY
LLPC, LLPY
LLPA, LLPX
LLPB, LLPX
LLPB, LLPY
LLPC, LLPZ
LLPC, LLPY

Figure 24: Printed material provided to the participants. Left: Scheduled tasks for the flights assigned to the pilot. Right: Chart of LLPs.

that task and, as mentioned, the level of attention was evaluated by observation. Fig. 23 bottom, shows an example of those values as seen by the Supervisor in the workload timeline. For example, the change of the C3 link, which should not be delayed to avoid losing the aircraft, received a value of 3 out of 5. It could be argued that such an important task should receive a higher value. The reason is that the simple input element in the interface, with the preview that clearly shows the availability of the frequency, allows a quick decision and execution, in a matter of seconds. If the planned frequency has an unexpected quality of 0, the impact is suffered mainly in the time for execution because the pilot will check other frequencies to select an alternative with higher quality, a trial-and-error process which does not require a high cognitive process. However, an increase in stress could appear if the task is started too close or passing the waypoint, the planned frequency is not available and the alternative has to be quickly found to avoid consuming the buffer area. This kind of situations, not inherent to the tasks, are not considered while allocating resources in the workload profile; it is assumed that tasks are executed with appropriate timing. Anyway, the worst-case selected strategy is expected to provide some margin for these events. In the end, this is an example of how the usability of the GCS has an influence over the time and level of dedication required by the task. The LLP change was also weighted with a 3, which could be considered too high for a non-urgent task. The reason behind this is that, while using the alternative plans chart, the pilot moves all his/her attention to the booklet, which is a relatively slow action; after that, refocusing on the monitor and resuming a picture of the situation seems to require extra mental effort. The *direct to* clearance received a 4 because it contains demanding and pressing actions, particularly having to check the link availability and reply to the message before exhausting the air system timeout. Tasks requiring complete and immediate dedication like contingencies, or the control migration, especially for the receiving pilot, would receive a 5. The default value for any flight when it only requires eventual monitoring is 1, to reflect that the very fact of performing concurrent tasks, even when they do not consume any relevant resource at a given moment, adds extra workload; this is known as the *cost of concurrence* [116].

It should be stressed that the tasks were deliberately ballasted to recreate cognitive processes of clearly different tasks. The printed LLP chart could be easily embedded in the interface and show the pilot the alternative plans considering the aircraft status, but the aim was to simulate a situation where the pilot is focused for a while making a decision, while keeping in mind a picture of the situation of the rest of flights and the time left to execute any next or pending action. Guidelines like reminders and monitoring were not available in this stage. The reason is that the scenarios were designed to identify problematic levels of workload and task overlapping, check the pilot's workload management strategies or detect any usability weakness or aid in these stressful situations. In this way, the workload threshold could be determined and incompatibilities in task overlapping detected. Evaluating the workload without the envisioned aids follows the policy that pilots should not rely on reminders and warnings. The system, acting just as a helper to avoid any dangerous consequences, does not try to avoid proactive awareness.

Besides the maximum total workload, other exclusion criteria can be specified for the Scheduler to avoid undesirable situations if only the quantitative limit is considered. For instance, in the experiments, it was set a maximum of six flights per pilot, no more than one handover at once or a maximum of two concurrent link changes. While the limit of workload was never exceeded during the pre-flight scheduling, it would be surpassed at some points during the exercise as a result of non-scheduled events: the ATC requests.

Four participants performed two scenarios, each one including four aircraft during the en route flight stage, with a duration of around 12 minutes each. Scenarios were deliberately loaded with stressing overlaps of tasks to evaluate the more interesting worst cases. The tasks, sometimes placed very close together or overlapping each other, were the following, distributed throughout the four flights:

- *Scenario 1*. Two changes of secondary frequency; one change of both primary and secondary frequencies; three more frequency changes with the planned link unavailable; three direct to clearances, just one of them feasible, and four LLP change requests.
- *Scenario 2*. Two changes of secondary frequency; four frequency changes with the expected frequency unavailable; six direct to clearances, just three of them feasible; two LLP change requests.

Six participants of a range of ages between 18 and 60 years, university graduates with advanced experience using computers were trained about the needs of the simulated RPAS, the CPDLC, and how to use the interface following a tutorial with guided exercises. After the training, each of them piloted up to four scenarios with a single aircraft each, where the tasks appeared sufficiently separated so as not to influence each other. That served to measure the average resources required to perform each task.

Results

Following are the observations of the performance under the high level of workload recreated in the multi-RPAS scenarios of this stage.

TIME TO REPLY TO MESSAGES FROM THE CONTROLLER In the results, the response time to the request for an LLP change and to the *direct to* clearance were separated because the first one does not require an evaluation (was always WILCO), but the *direct to* required checking if there was a link available for the shortcut first. Table 1 shows the values of the pilot response delay in three different situations: the message arrives while monitoring a single aircraft, while monitoring four, or while monitoring three and performing a task in a fourth. Only a couple of measurements could be registered for a case of having four aircraft and two overlapped tasks in different flights when the message arrives and the mean was 36.5 s.

TASK AMPLITUDES An excess of the *pilot execution delay* can also affect the ATC [78], that is why the Workload Profiler will reject any task overlap that could increase it beyond some limit. This limit, which can be seen represented in (Fig. 8), should be agreed and could depend on the final RPAS integration if RPAS are

Table 1: Time to reply to ATC messages in seconds

1 flight	**Mean (samples)**	**Median**	**Mode**	**σ**
LLP change	40.9 (10)	37.5	73	29.82
Direct to	51.8 (5)	51	-	33.39
4 flights no current tasks				
LLP change	18.5 (8)	12.5	9	11.03
Direct to	15.82 (17)	14	8	9.79
4 flights one current tasks				
LLP change	22.33 (12)	11.5	5, 8	22.58
Direct to	22.63 (8)	13	-	17.72

Table 2: Task amplitudes in seconds by number of flights and overlapped tasks

1 flight		**Mean (samples)**	**Median**	**Mode**	**σ**
LLP change		58.9 (10)	57.5	73	23.23
Direct to		99.8 (5)	103	-	51.66
C3 link change	pre	102.2 (18)	82.5	47	64.91
	post	5.57 (14)	4.5	2	5.36
4 flights					
C3 link change	pre	83.13 (8)	83.5	91	75.8
	post	2.5 (8)	1	0	3.25
2 C3 link changes	pre	41.08 (12)	38	67	29.90
	post	2.92 (12)	0	0	4.23
2 LLP changes & 1 C3 link change	LLP	82 (4)	76.5	-	50.8
	C3 pre	43.5 (2)	43.5	-	0.7
	C3 post	5.5	5.5	-	4.95

allowed some margin of tolerance with respect to manned aircraft considering the unavoidable latency. Table 2 reflects the impact of the task overlaps in the task duration (which includes the *pilot execution delay*).

ERRORS, OMISSIONS As there was not too much time available to train the participants, most errors had to do with their low experience with the interface and procedures, and these disappeared or decreased with practice. More interesting issues are those related to multi-piloting under stressful situations and are described below.

i. Doing the last action of a task, a participant receives a *direct to* clearance. He decides to finish the current task before sending the reply, contradicting the indications to this respect. While doing so another message arrives. He finishes the first task, attends to the third message and completely forgets to reply to the second, getting a timeout. The top workload here was 12; the pre-flight scheduling threshold was 10.

ii. A pilot was executing an LLP change request when a same request arrives for a different flight, to which he replies with a STANDBY before coming back to finish the previous task. This was unnecessary, as they were told that the answer would always be WILCO. He updates the first LLP but then realises that a link change is very close and starts to check the availability of the scheduled frequencies. He will forget to send the notification for the first LLP change to the controller, while the second message becomes a timeout. This stresses him so much that he fails with the link update because he was checking the link quality in the wrong segment; the aircraft exhausts the buffer area and loses the link. The maximum workload weight during this overlap was 17.

iii. With no other pending task, a user did not realise that one aircraft was reaching a waypoint where a change of link was planned and loses the aircraft.

iv. A participant receives an LLP change while there is no other ongoing task, applies the change correctly, but forgets to communicate the new plan to the ATC. At this moment, a couple of aircraft were close to reaching waypoints with scheduled changes of frequency. In this same point of the scenario, another pilot replied WILCO to the instruction but never applied the change of LLP. A third one even failed to reply to the message on time. The expected workload value when the instruction reaches them was 14, but it became 20 very soon, due to the pair of aircraft reaching the scheduled change of frequency. The reason why they left the LLP unfinished was probably that, during the training, they were told to give preference to the more urgent tasks. In the end, an LLP is something that would be executed in case the link is lost; but if the frequency is not changed when required, probably the link will be lost. So maybe they decided to attend to the other flights first and then just forgot to finish the LLP task.

v. In the previous situation, 70 s after the LLP change request and about 60 s before the two concurrent changes of frequency, a *direct to* clearance arrives generating a peak of workload of 18 units. Three of the pilots had previously lost that flight, so just one of them saw this message and was unable even to reply on time; he was too busy checking the availability of the frequencies and even forgot the STANDBY option. He failed to update the frequency of one of the aircraft.

Discussion

The observation of the performance of the pilots during the experiments, made it possible to check live the statements of previous research about the engagement of the pilots or the strategies to handle the workload, and confirmed the suitability of the suggestions for remote pilot stations found in [20] as a real need to improve performance and avoid errors. Even when managing several aircraft, when they could see that the closest task was some time ahead, you could see them bored, distracted, lost in thought, or looking at their smartphones. They usually reacted like this when the expected idle time ahead was around two minutes or greater. This sheds some light about the monotony of monitoring just one aircraft.

Regarding the time to answer the controllers, it was between 15 and 51 s, depending on whether the message required previous evaluation of the answer or there were parallel tasks. This is far from tolerable in a real scenario. In the manned case and voice communications the average time since the controller starts talking until the pilot finishes the read-back was found to be about 11 s in [117] (in an area of high density of traffic). [78] measured a mean of 2.5 s in unmanned aircraft simulations with a single flight, voice communications and no parallel tasks (contrary to the previous value, this does not include the readback). An added delay of 5 s is hardly accepted by the ATC [78]; then, the best case measured here, 15 s, would be somewhat accepted. But it must be considered that the present experiments were performed by non-professional pseudo-pilots with few opportunities to practise. Also, the workload level reached many times would not be allowed by the system in practice. However, pilots' response and execution time visibly improved with practice, then a clear decrease in the response time could be expected in more realistic conditions. In any case, some measures were later implemented to help pilots to reduce this mark, like automated replies, reading aloud background messages or the possibility of replying using voice recognition.

Although it could not be inferred from the log, the observation of some of the participants showed that when the CPDLC air system timer expired it was usually because the pilot forgot to answer the message, not because he/she was doing other tasks. It made clear that the visual icon notifying an open message was not enough the way it was, and it inspired a better way to highlight it. Only a couple of participants used the STANDBY message, which adds 100 s more to the timer. One of the participants, when asked about what she thought of the fact that the response time was significantly shorter when she had several aircraft, said that she was quite a lot more engaged having the workload of the four flights scenario than with the boring exercises with just one aircraft, where the connections/disconnections of attention where sparser; with four flights she had to attend to the messages quickly to keep paying attention to the rest of the flights. That, which corresponds to the Yerkes-Dodson law, should be considered the reason why the average time amplitude of the task execution and the pilot communication delay decreases with several flights. It reflects the pilots' strategies to manage their resources to maintain the level of performance, something that has already been analysed for the specific multi-UAV case [118] and the ATC case [119]. Experience is related to the use of good strategies to manage the workload [120] and probably some task overlaps that led to errors could have been solved successfully with more training, or showing the pilots strategies to interleave the actions of parallel tasks to avoid timeouts and loss of link.

From the results, it seems that the intuitive maximum workload threshold of 10 during the scheduling of planned tasks was not misguided; errors begin to appear when the events raise the sum of workload weight to 12. In the following set of experiments, the threshold was lowered to 6 to avoid being easily overpassed when ATC instructions arrived. However, in a final implementation, it would depend on each pilot's performance profile.

Among the problems arising during the overwhelming overlappings, i) showed the need to warn the pilot when the air system timeout (the time to reply to the message) was about to expire, and this feature was implemented. The iii) omission was addressed placing text reminders in the flight strip for close scheduled actions. To those instructions accepted but then partly or completely forgotten, the system would send a notification to the pilot showing the callsign and the pending action. These solutions can be seen in action in [43, screencast 9].

4.2 SECOND SET

Several months had passed from the first tests and participants requested the chance to refresh the concepts, procedures and usage of the interface, and could not remember the events of the scenarios. It did not seem likely that a possible improvement in performance could be just due to an increase in accumulated experience.

Design and Execution

Participants operated the same scenarios than those in the first set, this time with some of the aids, guidelines, warnings and monitorings implemented. These were expected to obviously contribute to reducing the number of errors, but it was an opportunity to check to what extent, and evaluate how appropriate was their how and when.

Some of the situations that in the first set of experiments led to surpass the workload threshold could be avoided in practice thanks to the Scheduler and the workload map learned by the Workload Module. An operative workload heatmap could have indicated a high probability of concurrent ATC instructions at some points, reducing the number of total flights or assisting the pilot somewhat. If not, another threshold to be considered is the one handled by the Workload Monitor, which would request the Event Handler to act when an excessive workload is detected for a pilot during the flight independently of that calculated from the ongoing tasks (therefore, the real stress experienced by the pilot). Reducing the threshold during the pre-flight scheduling also reduces the possibility to require the intervention of the system when these situations appear. Anyway, in this case, the decrease in the threshold did not reduce the number of flights, which was four in both scenarios, and the Workload Monitor was not implented. In summary, the results should be observed as related to a worst-case scenario, with peaks of workload that would probably not be allowed by the system in practice, but able to reveal the potential or the weaknesses of the features and design decisions. The aids included in this set of experiments were the following:

- *Link update reminders*
  The link changes were scheduled at specific waypoints representing the close end of coverage for the current link, or the starting of an area with better quality using a different frequency. The most delicate case of coverage transition would be that represented in Fig. 25. If the pilot updates the link in

1, where there is no coverage for the next frequency yet, there will be a link loss, and after the period of time specified by the lost link timeout value, the onboard computer will execute the LLP loaded, which could negatively affect the traffic and the task of the controllers. The link would also be lost if the pilot exhausts the buffer area without changing the link. To avoid these mishaps a couple of warnings were implemented. One of them writes the words "LINK CHANGE" in blue in the strip when a scheduled link change is 30 s ahead (Fig. 18, 1). Participants were told that this was a reminder, not necessarily an appropriate moment to change the link, as it depends on the lost link timeout (Fig. 25). They could use the time remaining to the next waypoint to decide the moment. If there is not coverage for the next frequency yet but the time remaining to reach to the area with coverage is lower than the LLT, the aircraft would recover the link before the LLP is loaded, so changing the link that soon would not have a bad impact. The second warning is triggered some seconds after passing the waypoint if the change has not been registered yet. The words "IN BUFFER AREA" are printed in red (Fig. 18, 2), and a synthesised voice reads them, preceded by the callsign. A screencast can be found in [43, Screencast 9].

- *Value checking*
  When receiving a clearance, if the pilot introduces a command with a value that does not match that specified in the instruction, such as an incorrect waypoint, speed or altitude, a request for confirmation showing the expected value is received before applying the command. If the pilot confirms anyway, the supervisor receives a notification and the controller could also receive a free text message notifying about this. A screencast is shown in [43, Screencast 8]. This kind of problems disappears when the instruction can be applied from the same message in the CPDLC display.

- *Unexpected actions*
  When an action that should be preceded by a clearance is detected, similarly as before, a confirmation is requested, and both the controller and supervisor are notified if confirmed ([43, Screencast 8]).

- *Monitoring of instructions*
  The system monitors the execution of the ATC requests only if these have been replied with a WILCO. If these must be immediately executed and they are not, the pilot receives a notification after a while. If still ignored, the supervisor is notified. For conditional clearances, the pilot is warned with a voice notification and a printed message in the strip when the condition is met; then the procedure is the same as the clearances of immediate application ([43, Screencast 8]).

- *Airsystem timeout warning*
  In CPDLC, messages have a limited time to be answered, after which pilots must ignore the content of the message and the controller reverts to voice to check the problem. To avoid overburdening the ATC, a synthesised voice warns the pilot when there are 30 s left to answer. A screencast is shown in [43, Screencast 9].

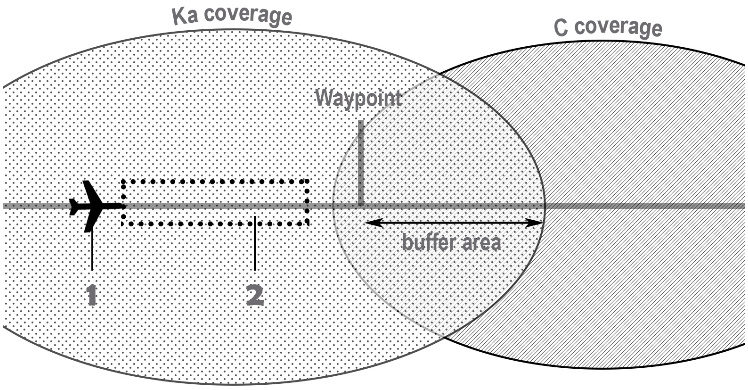


Figure 25: The dotted rectangle represents the distance ahead given the current speed and the lost link timeout. If the pilot changes the link to C in 1, which does not have coverage yet, the lost link procedure would be executed when reaching the right end of the box. If they need to change the link before the scheduled waypoint to attend concurrent tasks, they should do it in a position like 2, from where the distance to timeout would overlap the C coverage and the link would be recovered before it is exhausted. The interface shows both the values of the lost link timeout and the seconds remaining to reach the waypoint; then the pilots do not need do mentally calculate the distance, just checking that the first value is longer than the second value.

Results and Discussion

Following are the observations on the result of putting to the test some of the measures of usability and safety conceived and the comparison of the pilots' performance with that showed when the control of the flights only depended on their ability to keep in mind a picture of the situation of every flight.

The voice reading the messages received by background flights was considered useful by a couple of participants, who paid attention to the content of the message and the callsign. The other two found the voice useful as an additional highlight of an unseen message (an envelope icon is displayed too), but felt too stressed and focused on their current tasks to pay attention to what the voice was saying.

The "LINK CHANGE" print, conceived to be useful in cases of loss of awareness due to low activity, was instead considered especially useful in moments of high activity and overlapping of tasks. A couple of participants suggested, however, that it should be accompanied by a voice warning, a sound, or blinking. Suggestions like this need to be considered carefully. The aim behind the quantity and kind of warnings is to find a middle ground where these aids help avoid problems but do not encourage a loss of awareness. Too much guiding could induce pilots to let themselves be carried along by reminders instead of actively keeping in mind the situation of the flight, pending actions or close tasks. Even worse, an annoying excess of notifications is an easy way to get people to ignore them.

The "IN BUFFER AREA" print was, however, considered redundant; participants said that the previous warning already put them on alert. The aim of the "LINK CHANGE" reminder was to avoid the change of link in the nick of time;

Table 3: Link update done before/on the waypoint (total changes) - errors

Participant	Without reminder	With reminder
1	7 (9) - 1	9 (11) - 0
2	8 (9) - 4	10 (11) - 1
3	8 (9) - 3	10 (12) - 0
4	9 (9) - 0	9 (11) - 1

Table 4: Severity of errors without and with monitoring and guiding

Pilot	1st tests	2nd tests
1	2× forgets LLP change report 1× CPDLC timeout 3× lost link	1× wrong LLP letter in report 2 × CPDLC timeout 1 × lost link
2	3 × lost link	No errors
3	1 × applies instruction without WILCO 4 × CPDLC timeout 2 × lost link	1× wrong LLP letter in report 1 × forgets setting LLP but not the report
4	1 × forgets LLP change report 1 × forgets LLP change	2 × wrong LLP letter in report 1 × LLP change to wrong aircraft

ideally, it should be applied some seconds before/after overflying the waypoint. Table 3 compares when the link update was done depending on whether they had the reminder or not. On it, the change of link is considered done past the scheduled waypoint if it is forgotten or done still inside the buffer area but after at least ten seconds once the waypoint is passed. The changes previous or over the waypoint include the incorrect ones, as this only evaluates the awareness of the update. The values in parentheses are the total link changes in both scenarios. The number of times that the pilot changes the link in a timely manner show no remarkable difference having a reminder, but there is a clear decrease in the number of errors when the link change is applied when the reminder is printed. Maybe the reminder motivated them to check sooner which was the new frequency and its availability, or it pressed them to finish their current task, or just helped them to better organise their time and this allowed them to check their steps more carefully.

Another interesting decrease observed was that errors of the same participants did reduce in quantity but, what is most important, in severity. Table 4 details the kind of errors before and after the implemented aids. This was only after a few added opportunities to practice, so these results could be expected to improve much better with experience.

The CPDLC air system timeout aural warning [43, Screencast 9] was not considered useful by a couple of participants, but in this respect, it is important to stress that both of them failed to comply with the priority rules, so they probably did not understand their importance very well. Participants were told that the highest priority was, from more to less urgent: keeping the link alive, answering the controller, execution of instructions and finally the change of LLP. However, while carrying out an LLP change request, they received a message in the background and preferred to finish what they were doing instead of answering the other controller; they preferred to get rid of the current task first. One of them arrived on time to reply, the other one exhausted the timeout. In one of the cases, the synthesised voice read a request for an LLP change in the background. Participants were told to always answer with a WILCO, as there would always be an alternative plan available. Selecting the other flight, then the message received in the CPDLC display, pressing the WILCO button and then coming back to the previous flight, was all a matter of four clicks and can be solved in two seconds without involving any decision making, but this participant preferred to try to finish the current task first, failing to reply on time. The second participant received a *direct to* clearance while doing a task. To reply, first, they need to check if there is link coverage available for the new segment, which is slightly more demanding and could discourage the pilot from stopping what she was doing to reply. While they still have the option to send a STANDBY, which is also a matter of two seconds, to double the time allowed to reply, this pilot still preferred to take a risk and finish her current task, and ended up replying in extremis. The act of postponing an incoming activity while busy on one involving a high cognitive process seems to be a default in human behaviour, as shown in [121], while we can expect a task switch if the current task does not involve a high cognitive workload. In the couple of cases mentioned, the task they were performing was the most deferrable of all (in the second case, the LLP change as part of a *direct to*), but it seems that what made them reluctant to answer was facing the flight swap; having to load in your mind the situation of the other flight and then come back to the current one. So, in this respect, the interface should address a couple of issues. First, trying to reduce the involvement of cognitive resources in the *mental loading* of the second flight situation. Maybe the possibility to reply to background messages using voice recognition (partially implemented but not used during the experiments) to avoid the overload of visual resources, could alleviate some of these situations, a *task interference* solution as shown in [122][13][123] and with a promising performance in [124]. Second, tasks with higher priority should not be deferred in favour of those with lower priority, whatever the mental workload each of them involves. It would be interesting to try to highlight this fact in the interface, warning pilots to switch and respect the priority rules if they let themselves get carried away by the inclination to clear a task from their minds.

The notifications of wrong values introduced and the request for confirmation of unexpected actions came into play a few times, as sometimes participants forgot to select the correct flight before applying a command, or because they forgot to accept the instruction before applying, so they found it especially useful. Two participants suggested an added measure: a warning before applying a change of link frequency that would mean a loss of link. This measure was initially considered but left out because in the exercises they were allowed to do it if the aircraft reached the area of coverage before the lost link timeout was exhausted (Fig. 25) and, being a common and voluntary action in the scenarios, it was thought that a warning could be annoying. But one participant emphasised that she preferred to assume that cost and be sure that she would never lose the link by mistake, so it would probably be a good candidate for a customizable warning. Anyway, a request for confirmation seems appropriate for this kind of situations. [55] describes two main ways to cope with unexpected values at a GCS level: warn or ban. The problem with the first option is that during high levels of workload, the warnings were unnoticed; the second option was rejected by pilots. NPI could ask for confirmation: the action is not forbidden, but the pilot is forced to confirm and therefore pay attention to the warning to go on.

One particular mistake appeared in every tester related to the request for an LLP change, reinforcing the importance of automating the reports as much as possible to avoid input errors. They were told that there was always an alternative plan, so the reply should always be a WILCO. After that, they would check the substitute procedure on a paper chart given the position, altitude and remaining fuel. Once set, they should find time to inform the controller of the new procedure (identified as LLPA, LLPB. . .) using a free text message. But all of them sent, at least once, the wrong letter of the new plan, although correctly set in the aircraft. One participant sent the report without previously setting the plan, nor doing it later. The inputs monitored rely on the syntax of the message elements to parse and compare the values (see Fig. 27). In a free text message element, the report could come like *"LLPlan set to X"*, *"LLPY set"* or *"X changed to Y"*, which makes it difficult to compare the content with the plan just set. Of course, this report could have been considered automated from the beginning, as others are already for ATC report requests. The alternative procedure could also be suggested by the interface given the current status and readings, instead of using a printed chart. But the procedure, as designed, tried to represent a sample of demanding tasks in time and cognitive resources, where the pilot needs to take her/his eyes off the current screen for a moment. A real RPAS pilot even reported that this could take them up to five minutes. In practice, it seems appropriate to automate the report to the ATC once the LLP is changed to avoid these errors.

The lukewarm reception of the aural warnings, especially the synthesised voice, should be considered in perspective. In these experiments, the GCS was embedded in the NPI. This means that they are constantly looking at a unique screen, with all the information at a glance, so they can rely mainly on visual resources. But in a real installation, pilots could be focused on the GCS consisting of several monitors when an event appears in a background flight, and the fact of having a voice reading the message or describing the event, with no need to take their eyes from the current monitor, could be appreciated a lot more as this lets them know the urgency of the event and quickly decide their workload management strategy [123] to attend the new task. This would correspond to the findings of Wickens *et al.* [122] about the benefits of the cross-modal displays, also tested in flight simulation tasks, for better management of mental resources while attending multiple tasks.

Concerning the notifications (Fig. 16, 2), like those received when a pending action is delayed, users found it useful to remember which aircraft had the pending task, as these dialogues include the associated callsign in the tab. If they remember having a pending action but not in which flight, they are forced to click on each strip and check the CPDLC history or the printed scheduled tasks to find it out, which takes time. This revealed the need to print some kind of pending action icon in the strip. It should be different depending on whether this comes from an ATC request or a scheduled task, so they could know beforehand where to check the details of the action. Same need was observed for open messages. By now, an open message icon is printed on the CPDLC display next to the message that has arrived to remark that it has not been answered. But you need to have that aircraft selected to see the icon. When a new message in a background flight arrives, an envelope icon, differently coloured depending on the severity, is added to the strip, meaning that there is an unread message. This icon disappears when that aircraft is selected. If the pilot does not answer the message at that moment, and moves to another flight, there are no cues in the strip to remember which was the callsign with the open message, and pilots looked for it by trial and error.

These observations seem to satisfy and get the benefits of Endsley's recommendations [73] for each level of situation awareness. For the first level, it guides the information sampling by the visual (new message icon, variable highlight or the open message icon that should be added to the strip) and aural clues (a new message sound differentiated by urgency and the short-time-left-to-answer warning). For the second level, it provides an intuitive and clear comprehension of the situation, reducing the requirements of working memory, which is key in multi-RPAS piloting, by the print of the previous communications, the dialogue colour grouping, horizontal alignment and open message icon in the history. The third level, the projections of future status, is addressed by the NPI in the strips (position in the segment, link quality, scheduled tasks).

4.3 OBSERVATIONS ON THE USE OF THE CPDLC DISPLAY

It is worth mentioning some aspects observed during the interaction of the participants with the CPDLC display. As previously mentioned, to answer a message, it must be selected first; this is the way in which the interface recognises that the user is not starting a new dialogue, or identifies which dialogue the reply belongs to if more than one is open at a given moment. The display tries to avoid sending a message element that should be associated with a previous one that was not selected, e. g. a WILCO with no previous uplink message selected. But a reply may use non-dependent elements, and then there is the possibility for the pilot to send a message with the intention to reply to another one that will not be closed, as it was not selected first. In those situations, they reported that the open message icon helped them to notice the error and notify the controller.

The message element selection and composition proved to be a complete success. To reduce the complexity of the experiments, the kind of messages they were required to send and reply to was very small, all single-element. Each of their actions was registered with a timestamp, and one of them was able to search and find the element (using the search by code), fill in its value and send it in a record of 2 s. While such an extreme mark is not a goal, it is an example that, with the suggested display, the head-down is not the main problem anymore.

The following observations are specific to the multi-RPAS case. For instance, the voice warning when there are 30 s left to answer an open message was considered partially useful. Participants said that it helped them to know that there was an open message, but not in which flight. The reason seems to be that the voice begins the warning reading the callsign, but following the convention of reading it as the airline name plus the last numbers (BAW962 as "Speedbird nine six two"). As they were not used to such phraseology, and could not associate the ICAO airline designator with the airline name, they were told to pay attention just to the numbers. But that is not easy, particularly while being busy on other tasks, so they ended up searching the open message through the flight list by trial and error. This showed again the need to place an open message icon in the strip, which could be useful even for experienced pilots, to avoid the need to remember the associated callsign.

It was mentioned that when a message is received in a background flight, the synthesised voice reads its content. One reason is to provide a clue about the content, to allow the pilot to begin to think about the decision while moving attention to the NPI and selecting that flight to compose and send the reply. Another reason, not used during the experiments, is to allow the pilot to verbally compose and send the reply. Participants reported not paying attention to the reading of the message if they were very focused on another task; but that it was nevertheless a helpful redundant way to highlight that there was a new message (an envelope icon is also displayed in the strip with a colour related to the urgency).

To perform the experiments, participants had to be trained in concepts associated with the operation of an RPAS, and remember the procedures designed for the experiments. They had to make decisions and manage the workload, follow priorities, accomplish scheduled tasks and attend to ATC requirements. In such demanding exercises, the use of the display did not appear as a contributor to cognitive workload. They easily associated it with a chatroom with the controller, a chat with predefined text and rules, and soon used it confidently, trying to memorise the element code to quickly select it in the search box, and with minor errors only appearing during moments of high workload.

With no previous recommendations in this regard, they took the precaution to take a look at the display, before and after introducing a command — the Apply button was disabled to generate workload — to check that they had introduced the correct value (in the second set of experiments, notifications for those kinds of errors were enabled, but they were not warned about it). And, periodically, on their own initiative, they checked the history and contrasted it with the status of the flight to be sure that all the actions requested by the controllers had been executed (also, they were not warned that the system was monitoring the accomplishment of the execution), something that they could contrast without any action required, just selecting the flight, as the readings and status were next to the message history. This reveals how appropriate it is to provide the pilot with a print of the log of the communications, not only of the last message; this history should always be visible at a glance, with no need to navigate menus to check each message. A possible improvement would be to somehow mark those instructions that have been executed, so the pilot could check them even more quickly. It is currently done when using the Apply button, but not when the instruction is done using the GCS.

5 CONCLUSIONS

The ConOps proposed here pursues the feasibility, from a human factors perspective, of having a single pilot/aircrew controlling several RPASs concurrently in non-segregated airspace. The concept seeks a safe integration of these operations by providing pilots with tools to better manage the workload and keep the awareness, and the backup of a system able to monitor and support this kind of operations. As any impact on the work of the ATC due to delays or errors as a consequence of this concurrent piloting must be avoided, a set of measures at several levels is suggested, which include workload prediction and balance, pilot activity monitoring and a special emphasis on interface usability. The concept relies greatly on the exploitation of the potential of Controller-Pilot Data Link Communications, anticipating future widespread implementation and full use.

The workload management includes a scheduler responsible for suggesting an assignment of the flights in a way that avoids the overlap of excessive workload. And during the flight, the update of the forecast of workload is constantly done to check if there is a need to apply any measure to avoid the pilot facing excessive tasks. Besides that monitoring, completed and incompleted actions are tracked to warn the pilot or the supervisor in case unexpected situations or errors are detected. To illustrate some of these features, test them and identify more needs, a prototype was implemented, a description of the workload based on observation was determined and some experiments performed. The scenarios simulated stressful situations to check when and why the pilots failed, and the impact on the time to reply to the ATC and execute the instructions. Those situations shed light on the level of maximum workload that should be allowed, and the improvements on the usability that should be addressed, which were partially implemented. The subsequent experiments, with some of the envisioned features implemented, showed a decrease in the number of errors, oversights and subjective stress.

If all the features and tools suggested are finally implemented, put to the test and pilots conclude that they do not feel like they are able to safely control more than one aircraft, their suggestions and opinions could be introduced in a new iteration of the design and development process, eventually leading to a version that could satisfy their needs. But as a whole, it is not expected that the current framework could be rejected because it failed in adding value. Therefore, its success should be measured in other parameters. One could be the total number of flights that a pilot could control. That would depend a lot of the kind of missions. If we consider the case of High Altitude Long Endurance (HALE) or High Altitude Pseudo Satellite (HAPS) aircraft providing communication access, probably the pilot will be able to handle several of them. Other kinds of missions like those with close coordination with the payload operators could not allow such level of concurrency. In the worst case, if the number of aircraft was reduced to two, the operator is still doubling the productivity of the pilot workday. Even if the scheduling determines that there is one flight that is incompatible to be controlled in parallel with others, in that case, this operation could leverage the safety measures provided by the system, like the monitorings, warnings and reminders.

Some pieces of the ConOps could offer collateral benefits by themselves, even separated from the system. The implementation of the standardised CPDLC display could serve, by itself, as a powerful tool to train pilots and controllers, who will need to gradually learn and maintain familiarity with the procedures and the message elements that will be used whether they are manned or unmanned aircraft, a training that is requested in the implementations guidelines of the data link standard ([36] Part I, Ch. 3, App. B, 2.4 and 2.5). As reported in [88], in the U.S., air carriers mainly refer pilots to the Flight Operations Manual for the CPDLC training, and frequently they do not have the chance to practise simulations before the flight, although recommended by the Federal Aviation Administration (FAA) [125, Chapter 8.3]. It could also be used by researchers to perform faithful simulations and check the potential problems that could arise in practice from procedures and specific message elements [87]. The NtoM CPDLC display allows scripts to be run simulating the role of the ATC, as it was done for the experiments; once different exercises and scenarios are designed, pilots could practise by themselves with no need of a second person acting as a controller. Clients and the server connect to each other using the Data Distribution Service (DDS) standard, which allows the simulation of different scenarios of QoS. In this way, users can face different levels of communication reliability and train in the Standard Operating Procedures (SOPs) established for the problems arising from connectivity issues. As an isolated module, it could be used together with any existing flight simulator and ATC training tool — it requires about a third of the width of a 23″ monitor. In any event, the present display does not try to mimic the current cockpit displays as other training tools already do [126]; then, ultimately, manned aircraft pilots should also practise the menu navigation and behaviour of the specific cockpit units to be used. Hopefully, considering the effort put on its usability and the decrease of the head down it could provide, it could serve as inspiration for the implementation of these displays in future GCS's, as it has been conceived to leverage the potential and flexibility of a mature CPDLC implementation. This applies also for the ATC case; a controller surveyed about the display found it quite complete, and he liked the fact that the elements could be quickly found and selected but also that it was not a saturated environment of lists and information, which would difficult the operation. He found the message history as a chat room as more easy and intuitive and highlighted that it is very necessary to have the communication history at hand to check the correctness of the previous instructions delivered, noting that the current system used by ENAIRE also provided a plain text list of the history, but this is opened by selecting the tag in the radar display and then a button inside its contextual dialogue.

The system is able to compare the pilot's actions with the flight plan and the controller's instructions, which is useful as a contingency detection measure even if we are considering a one-to-one pilot-aircraft relationship. The measure suggested of automating the report of unexpected actions using CPDLC to provide the details of the abnormal command was celebrated by the controllers surveyed, even when it could be only a redundant measure to back up their monitoring system or the use of ADS-C reports of events.

With the transfer of control handled by the system, any error related to the misconfiguration of the receiving GCS disappears. The whole process and the information involved, like GCS modes, aircraft status, pending actions and communications, are kept and transferred by the server, so the role of the transferring pilot is minimum or even nonexistent. This is especially interesting when the transfer was requested as a consequence of an excessive workload, due to GCS malfunctioning or pilot incapacitation. To make it possible, we need a GCS providing an API for the NPI to be able to apply commands, load/unload flights and listen to the pilot activity and flight instruments readings. One of the measures of this process included a couple of informative messages to the CDA about the beginning and end of the migration procedure. That way, if the controller was about to send an instruction to the aircraft, he/she has the option to find an alternative or wait a bit, because probably the response and/or execution could be delayed. While this measure contradicts the guidelines about RPAS integration, which determines that these aircraft's idiosyncrasy must be transparent from the ATC perspective and controllers should not be overloaded with extra requirements, it was considered justified and helpful. Asking a couple of controllers about the suitability of this measure, they agreed that it would be useful as long as it would not mean an increase of the communications — migrations are not frequent during a flight but in a future the proportion of manned and unmanned aircraft could be very different —. To avoid this, one of them suggested that these messages could be reflected simply as a change of colour in the radar tag, instead of a message that needs to be read.

If monitoring or piloting the aircraft is considered to deserve a dedicated AVO per flight, the concept, or at least the workload scheduling, could still be applied to other roles, like dispatchers, first officers or payload operators.

Another piece of the system, this one with a long term benefit, would be the workload heatmap, which could provide useful information to different organisms and researchers, after the RPAS integration, to analyse the traffic management from the RPAS point of view, and apply corrective measures, if necessary, in the design of airways or procedures, conducive to better integration.

The balance of the workload could also help to avoid the burnout usually suffered by RPAS pilots [21], avoiding fatigue as an online planner and seeking a healthier shift distribution when acting as a workforce scheduler.

6 FUTURE WORK

During the development of the prototype, soon became clear that the initial goals regarding the implementation and the experiments was over-optimistic in time and resources. Some features let out would require long and complex experiments or are depended on the implementations of previous components, but others can be separately developed and included as they are completed, allowing different parallel research projects that could even be leveraged for other purposes than the present ConOps. Following is a list of the next works required to advance the prototype so that the thesis of the concept can be tested:

- *Workload Profiler*
  The experiments were performed by non-professional pseudo-pilots, with a minimalistic GCS and an estimation of the workload values based on observation. While more realistic testing with objective metrics would be desirable, the previous tests were intended to illustrate the input requirements of the scheduler: a workload model describing the cognitive load and time required for each pilot while performing each task or any task overlap. The generalization of this could be the focus of the next step of development, the description of a biometric agnostic methodology to build the personalised cognitive workload model from objective data and how to estimate each pilot workload threshold, i.e. to determine the algorithms underlying the Workload Profiler. With this model being dependent on the GCS usability, the procedures behind the tasks and the pilot's expertise, the methodology should address the propagation of the variation of any of those variables on the rest of the model. That way, the system will be able to automatically update it as pilots improve their skills or any task procedure changes or is automated in a particular platform.

- *Scheduler*
  With the Workload Profiler completed, the Scheduler will have one of its main inputs available, then it can be started the definition of the scheduling problem and implementation in a solver, for instance CPLEX [127].

- *Stages*
  Dependent on the implementation of the scheduler, these can be seen just as an added constraint, but integrating them affects the whole system. The airline operator (in practice the supervisor) needs a way to create them before and during the flight. This includes allowing the definition of the policies that apply when these are involved in some decision (e.g. should a balance of workload prevail over the preference defined by the stage?). The system must also monitor when the stage is about to start. The complexity escalates as more flexibility is allowed for their start and end points definition: at

a given time, at a certain distance from a waypoint, when a certain time elapsed since passing a waypoint, when overflying a state, etc.

- *Voice recognition*
  As aforementioned, the work on this feature was abandoned due to time limitations and lack of firsthand knowledge. But besides those technical issues, a careful usability study should guide the implementation, otherwise, a tool like this can easily go from comfortable and easy to annoying and complex. For that reason it should be designed and developed side by side with the pilots; not only to decide the *how* but also the *when*. Voice recognition and speech synthesis lose their meaning if they become more cognitively costly or more time consuming than using the visual resources of the graphic interface.

- *Workload Learner*
  Its development requires a well-advanced implementation of the system and a large set of experiments to capture the evolution of the skills. It includes the update of the workload profile of the pilot and the maximum workload threshold. The algorithms could rely on biometrics if the logistics are available, or in the performance of the pilot, provided by the GCS Monitor. In the first case, the Workload Profiler should be already finished, as it translates the measurements in the workload representation managed by the system.

- *CPDLC display*
  Despite the CPDLC procedures described here followed standard recommendations [29] [102], being a prototype, they do not try to be a reference of accuracy and completeness. The implementation does not cover the whole message set, variable formats, units and constraints. Although the greatest amount of effort was put into reflecting the specifications, no expert guidance was available to check the correctness of the interpretation of the documentation. Therefore subject matter experts revision is recommended prior to using this software as an educational tool. This lack of expert advice was also the reason why the input of the variables has not been constrained and the text boxes are not still error checked. Ideally, it should provide a unit selector and the error check would be dependant on the unit selected.

- *Workload Heatmap*
  The construction of this reference of workload contributors deserves per se its own branch of research. Its initialization requires determining which workload drivers can be inferred or leveraged from historic data and to which extent they contribute to the stress of the pilot. Then, once real measurements are gathered while pilots overfly the area, it must be decided how and which data are aggregated. Finally, an appropriate granularity in every dimension — which could not be necessarily homogeneous — should be established; able to provide a rich description but, at the same time avoiding an excessive cost in the access and update.

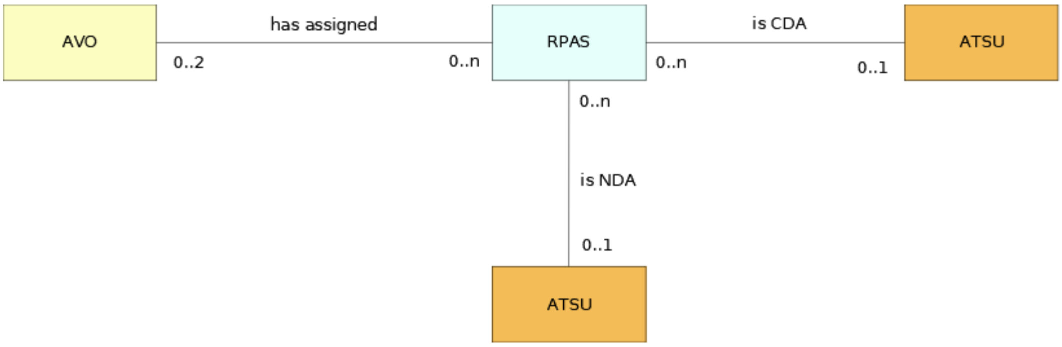


Figure 26: At a given moment, an AVO can have any quantity of flights assigned. An RPAS can have no pilot assigned if the flight definition has been introduced in the system but the scheduler still has not searched or found an available pilot before the flight; during the control migration the flight has two pilots assigned. The aircraft keeps at every moment a CDA assigned since the moment of the takeoff; as the control is transferred between ATSUs during the flight, the Next Data Authority is determined, which eventually becomes the CDA.

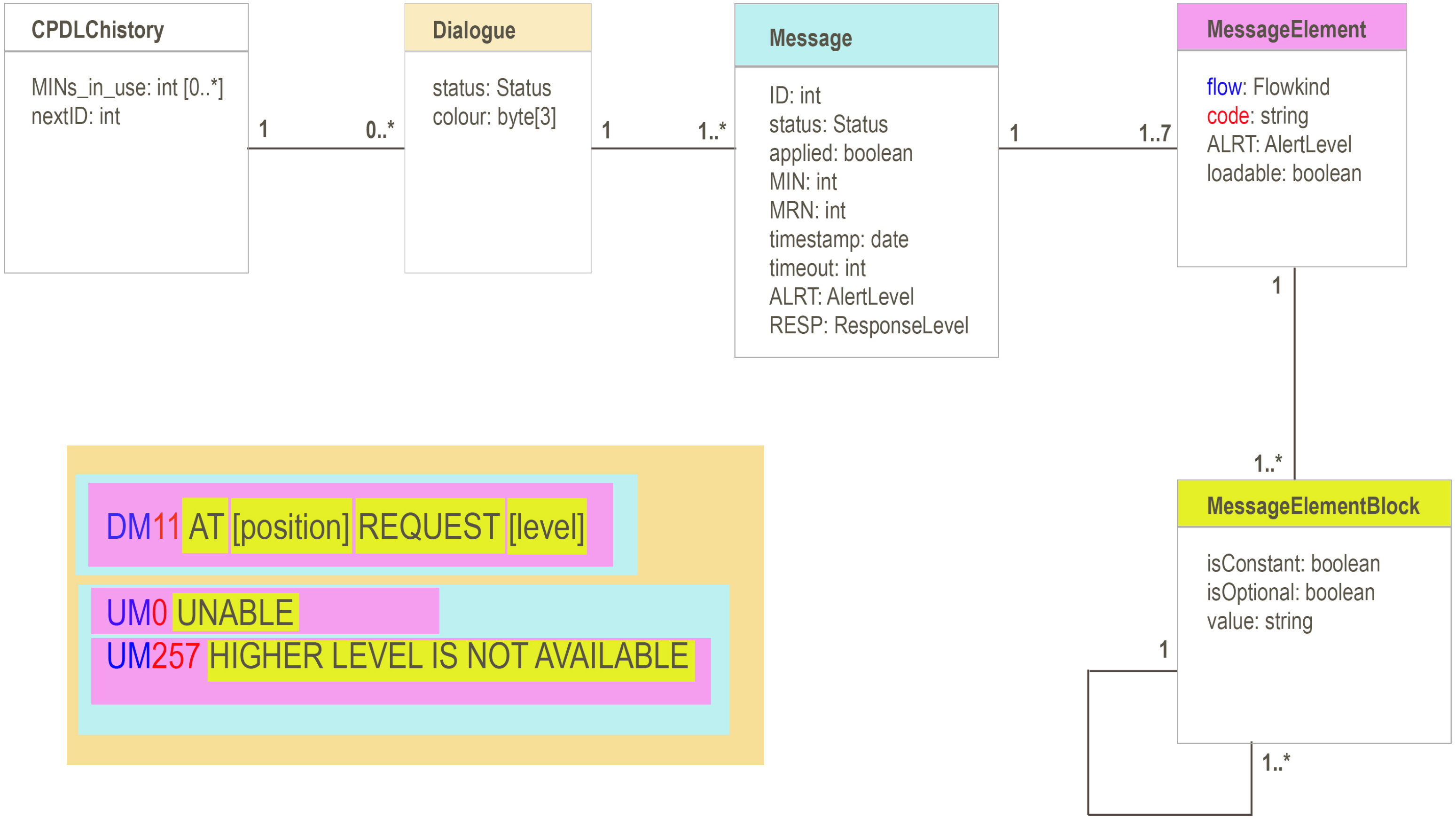


Figure 27: Diagram of classes representing the CPDLC messages and a coloured example of the the parts each class describes in a dialogue.

ANNEX B. WORKLOAD PROFILER

The Workload Profiler is a piece of the system that takes as input the measures of performance of the pilots while doing individual tasks and builds a worst-case estimation of the total cost when tasks are overlapped. This estimation is not a simple concatenation of the costs of the tasks involved; it is necessary to guarantee some constraints and the workload threshold of the pilot. The worst case has been chosen to respect the pilot's workload management strategy selected; the cost of the worst or more costly option cromprehends the resources needed by of the rest of strategies. This option is not usually the most optimal and easy, so pilots will probably tend to choose a smarter one. But the higher cost of the worst option provides some margin in case some of the actions are delayed for any reason, prevailing safety over productivity. Also, allocating resources to allow each pilot's preferred strategy or way of doing, as seen in previous works, plays in favour of their acceptance of the ConOps.

To illustrate some examples, a basic version of the Workload Profiler was modelled in Optimization Programming Language (OPL) and loaded in the IBM's CPLEX Optimizer [127]. A full operative model should be agreed with pilots and controllers to reflect the different constraints and preferences. For instance, to set priorities among tasks or actions. Following the description of the model, there is an example of the worst-but-feasible-case allocation of resources. When an improvement of the pilot skills doing some action is detected, that pilot's workload profile would be updated, propagating the recalculation of the new cost throughout all the overlaps containing the affected action.

Then, the goal is to find out if there are values meeting all the following constraints, and selecting the minor of them:

$$\sum_{i=1}^{n} \sum_{j=1}^{|T_i|} p^{ij} a_0^{ij} s(a^{ij}),$$

where

T_i : set of subtasks a^{ij} forming the task.

p^{ij} : priorities among subtasks. Penalises the total cost if the a less urgent subtask if placed before a more urgent one.

a_0^{ij} : point in time when the subtask starts.

$s(a^{ij})$: returns 1 if a^{ij} is the first subtask executed or the previous subtask does not belong to the same task. Otherwise, returns a constant value to penalize the combination. the aim is to give preference to those combinations with a change of flight after each task.

subject to:

$\forall < r^i, a^{i1} >$ in D_r, $a_0^{i1} \leqslant r^i + d_r^i$

$\forall < a^{ij}, a^{ij+1} >$ in D_b, $a_0^{ij+1} \leqslant a_0^{ij} + c^{ij} + d_b^{ij}$

$\forall a^{ij}, b^{pq}$ if $a_0^{ij} < b_0^{pq}$ then $b_0^{pq} \geqslant a_0^{ij} + c^{ij}$

$\forall a^{ij}, b^{pq}$ if $a_0^{ij} < b_0^{pq}$ and $i \neq p$ then $b_0^{pq} \geqslant a_0^{ij} + c^{ij} + s$

$< a^{ij}, a^{ij+1} >$ in P_i, $a_0^{ij+1} \geqslant a_0^{ij} + c^{ij}$.

where

r^i : task release times.

a^{ij} : subtask belonging to task i with an order j on it.

d_r^i : some tasks require a maximum delay between the release time and the starting of the first subtask, a value that can vary depending on the task and the situation, and is represented by this variable.

c^{ij} : period of time required for the subtask. Preemption at a task level but no job splitting considered at an subtask level.

d_b^{ij} : maximum delay allowed between action a^{ij} and a^{ij+1}.

s : swap time. Time associated to the cognitive cost of moving from one flight to another.

P : set of tuples describing the required order among subtasks inside a task.

D_r : set of tuples describing the tolerable delay since the task release moment.

D_b : set of tuples describing the tolerable delay among specific pairs of subtasks.

In natural lenguage and in the multi-RPAS context we could describe the previous objective function as follows. There are tasks related to different flights that overlap at a particular moment. These tasks do not necessarily start at the same time. Tasks are formed by different number and type of subtasks. While the subtasks inside a task do not need to be executed all of them to move to another task, and subtasks can be interleaved among different tasks, each subtask is assumed that will be fully completed before moving to another task. The objective function will look for the shortest feasible sequence of subtasks among tasks prioritising a) those sequences where the most urgent subtasks are executed before less urgent ones; and b) moving to a different task in another flight after finishing one subtask is there exists the possibility, which will depend on the unbalanced number of subtasks in the tasks.

The feasible quality of the sequence is achieved once all the constraints are satisfied. These determine that: i) subtasks can not overlap, ii) once a subtask is finished, moving to a different task adds a swap time; iii) some subtasks in a task must follow an order; iv) some pairs of subtasks can have restrictions on the maximum delay since one finishes and the following starts; and v) some tasks can have restrictions on the delay between the release of the tasks and the starting of its first action.

If there is a solution for the objective function, a following filter, not represented here, would be to check if the worload constraints are met. These would determine that the total workload along the sequence is kept below the pilot's threshold. Additional rules could determine the feasibility of each overlap, like kinds of tasks that cannot be overlapped in type or number.

The forecasts of the Workload Profiler would be used in different situations. If used before the flight, during the workforce scheduling, the tasks participating in the overlap would be those scheduled or those appearing with a high probability in the Workload Heatmap. If queried during the flights, the resources allocations would involve any kind of tasks, including those that cannot be predicted. In the first case, a detected overlap incompatibiity in the result allows to avoid a flight assignment, and in the second case, the system would be notified to provide assistance by automating tasks, transferring flights or applying whatever measure was available. The following illustrative example, while unrealistic, is useful to check how the time allocation is calculated involving different kinds of contraints. The unrealistic aspect comes from the fact that it is calculated as a pre-flight allocation while it contains ATC instructions that are not among those that could probably appear as expected in the Workload Heatmap:

Task 1. Link update. Scheduled change of C3 link.

- *Subtask 1. Link update.* Check frequency availability. In practice, there is not a fixed moment to begin this task. It should be done some time before the moment when the change of link is scheduled, but the exact moment is decided by the pilot. A worst situation is assumed: the pilot needs to find a different frequency from that initially planned.
- *Subtask 2. Link update.* Apply the change.

Task 2. Feasible direct to clearance.

- *Subtask 1. Read, evaluate and reply.* The evaluation of the frequencies indicates that executing the instruction is feasible.
- *Subtask 2. Execution.* Assuming no need to change the LLP. The execution should not be delayed more than 15 s since the WILCO is sent (value suggested by a controller for highly traffic congested areas).
- *Subtask 3. LLP update.*
  - *Action 1. Read, evaluate and reply.* Assumed that there always be an alternative plan, so pilots can automatically send a WILCO reply.
  - *Action 2. Load the alternative plan.*
  - *Action 3. Report the new plan loaded.*

In the example, shown in Table 5, the release times for the tasks are: Task 1, 0; Task 2, 10; Task 3, 20. However, we can see that the first action begins at 10, even then the Task 1 is released at 0. The reason is that the first subtask of Task 1 takes 42 units of time. The release time of the other tasks plus their tolerated delay to start (15) would end when Task 1 is being still executed, making the sequence not feasible. Obviously, that sequence solution is not a realistic one, which is not the goal, anyway. The pilot would start with the first task, probably

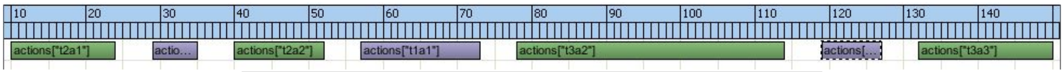
Table 5: Gantt chart of a solution obtained running the model described in CPLEX Optimization Studio IDE and tabular description of the same intervals sequence.

action	position	task	start	end	size
t2a1	0	1	10	23	13
t3a1	1	2	28	32	4
t2a2	2	1	37	41	4
t3a2	3	2	46	50	4
t1a1	4	0	55	97	42
t3a3	5	2	102	114	12
t1a2	6	0	119	122	3

without any clue that more tasks are about to arrive. If the Workload Profiler is called when an overlap appears during the flight, it should allow excluding from the calculations those subtasks already completed, or which is currently being attended. As desired, tasks appear interleaved and therefore with the time swap separation between subtasks. There was a delay constraint of 15 units between the end of the first subtask of Task 2 and its second one that is also satisfied.

Table 6 shows the same example using averaged subtask durations for a participant that later performed this scenario of overlappings, i.e. that is the estimation that this first version of the Workload Profiler would return for this pilot given the measurements of the pilot for the different tasks involved. Fig. 28 shows a comparison between the Workload Profiler time allocation and the real execution. Note that the top timeline does not try to provide an optimal strategy, but the longest feasible action sequence, being such feasibility the fact that all constraints appearing in the tasks can be satisfied. An example showing that the allocated sequence is not necessarily the most optimal can be seen in the link update. This task has not been subject to an intra-actions constraint because the pilot could apply the change at different moments depending on the situation. It could be updated before the scheduled point, after, still inside the buffer area, or even once passed the buffer area in case the change was only oriented to a QoS improvement and the link is not lost after the buffer. Not subject to any constraint about the update moment, the Workload Profiler places it quite a while after the scheduled point. However, the user waited to that moment to apply the change to avoid exhausting the buffer area. That is also the reason why the LLP change, not being and urgent action, was postponed. The starting time for the whole estimation was decided by taking the first action to arrive, which was the link update at a next waypoint, and checking the moment when the user usually starts to check the frequencies, that was about 1.5 min before reaching the scheduled waypoint. Even when the user started five seconds before that moment, the allocated total time cost still provides 15 s of margin with respect to the real execution.

Table 6: Estimated time allocation for real measured durations.



action	position	task	start	end	size
t2a1	0	1	10	24	14
t3a1	1	2	29	35	6
t2a2	2	1	40	52	12
t1a1	3	0	57	73	16
t3a2	4	2	78	114	36
t1a2	5	0	119	127	8
t3a3	6	2	132	150	18

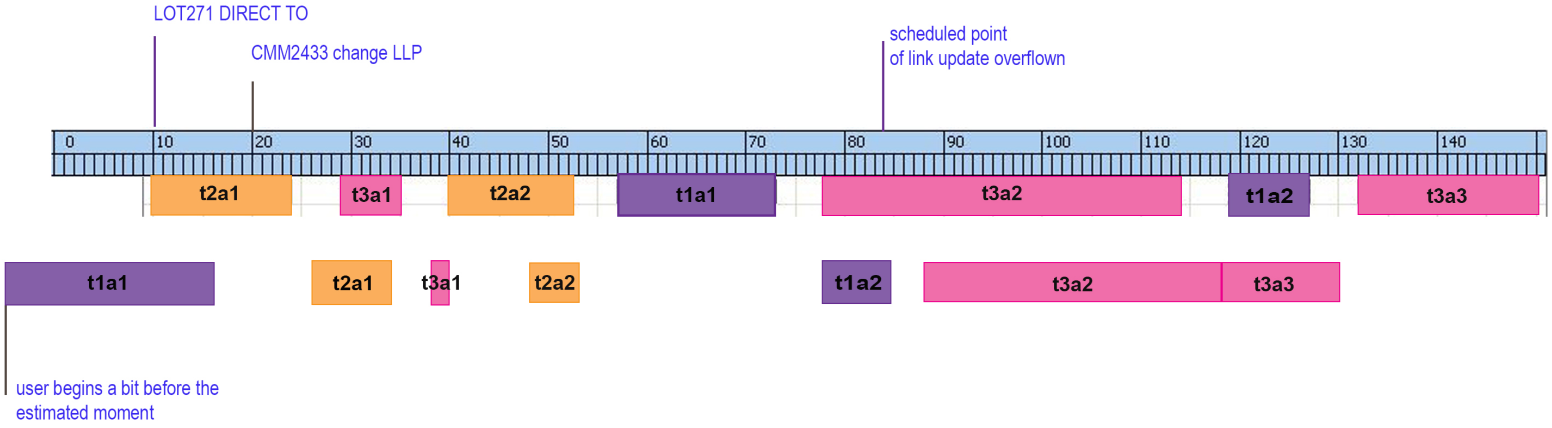


Figure 28: Comparison of time costs of the real execution timeline (bottom) and the allocated worst-case strategy (top).

PUBLICATIONS

- *NtoM: A Concept of Operations for Pilots of Multiple Remotely Piloted Aircraft*, International Review of Aerospace Engineering (IREASE), February 2019, open access. DOI: 10.15866/irease.v12i1.16153
- *Controller-Pilot Data Link Communications Display Oriented to Multiple Remotely Piloted Aircraft Pilots*, International Review of Aerospace Engineering (IREASE), February 2019, open access. DOI: 10.15866/irease.v12i1.15535
- *Dynamic Workload Management for Multi-RPAS Pilots*, International Review of Aerospace Engineering (IREASE), April 2019, open access. DOI: 10.15866/irease.v12i2.15334

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

[1] European Union. *Single European Sky*. 2015. URL: https://www.sesarju.eu/approach/objectives.

[2] European Union. *Functional Airspace Block*. 2015. URL: https://www.eurocontrol.int/functional-airspace-block-fabs-defragmenting-european-airspace.

[3] European Union. *Single European Sky ATM Research*. 2014. URL: http://ec.europa.eu/transport/modes/air/sesar/index_en.htm.

[4] European RPAS Steering Group (ERSG). *European RPAS Roadmap*. 2013. URL: http://ec.europa.eu/DocsRoom/documents/10484/attachments/1/translations/en/renditions/native.

[5] *European Drones Outlook Study*. Nov. 2016. URL: http://www.sesarju.eu/sites/default/files/documents/reports/European_Drones_Outlook_Study_2016.pdf.

[6] European Union. *Opening the aviation market to the civil use of remotely piloted aircraft systems in a safe and sustainable manner*. 2014. URL: http://ec.europa.eu/transport/modes/air/doc/com(2014)207_en.pdf.

[7] European Union. *IAI and Airbus Maritime Heron Completes 200 Flight Hours in Civilian European Airspace for FRONTEX*. 2018. URL: https://www.uasvision.com/2018/10/24/iai-and-airbus-maritime-heron-completed-200-flight-hours-in-civilian-european-airspace-for-frontex/.

[8] SESAR. *Concept of Operations for European UTM Systems (CORUS)*. 2018. URL: https://www.sesarju.eu/index.php/projects/corus.

[9] NASA. *Unmanned Aircraft System (UAS) Traffic Management (UTM)*. 2018. URL: https://utm.arc.nasa.gov/index.shtml.

[10] GUTMA. *Global UTM Association (GUTMA)*. 2018. URL: https://gutma.org/.

[11] *Drones as a leverage for jobs and new business opportunities*. Nov. 2016. URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjhlP612ZffAhWpRBUIHQJ5ADQQFjAAegQIABAC&url=https%3A%2F%2Fec.europa.eu%2Ftransport%2Fsites%2Ftransport%2Ffiles%2Fdrones-warsaw-declaration.pdf&usg=AOvVaw3dPvCqye7JQJNgY6pLpHAF.

[12] SESAR. *U-Space*. 2018. URL: https://www.sesarju.eu/U-space.

[13] Stephen R. Dixon and Christopher D. Wickens. “Control of Multiple-UAVs: A Workload Analysis”. In: *12th International Symposium on Aviation Psychology*. 2003. URL: http://www.dtic.mil/dtic/tr/fulltext/u2/a446844.pdf.

[14] H. A. Ruff et al. “Exploring Automation Issues in Supervisory Control of Multiple UAVs”. In: *Human Performance, Situation Awareness, and Automation Technology Conference*. 2004, pp. 218–222. URL: https://pdfs.semanticscholar.org/154f/e3bb05fc2f6d8b1bd7a1a4c5256e89e97ef1.pdf.

[15] C. C. Murray and W. Park. “Incorporating Human Factor Considerations in Unmanned Aerial Vehicle Routing”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43.4 (July 2013), pp. 860–874. ISSN: 2168-2216. DOI: 10.1109/TSMCA.2012.2216871.

[16] Lawrence A.M. Bush. “Shared UAV enterprise operator pooling framework (SUAVE): Chance constrained pooled fan-out queueing analysis”. In: *2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision, CogSIMA 2015*. 2015. ISBN: 9781479980154. DOI: 10.1109/COGSIMA.2015.7107970.

[17] M. L. Cummings et al. “Task Versus Vehicle-Based Control Paradigms in Multiple Unmanned Vehicle Supervision by a Single Operator”. In: *IEEE Transactions on Human-Machine Systems* 44.3 (June 2014), pp. 353–361. ISSN: 2168-2291. DOI: 10.1109/THMS.2014.2304962.

[18] Anthony P. Tvaryanas. *Human Factors Considerations in Migration of Unmanned Aircraft System (UAS) Operator Control*. Tech. rep. HSW-PE-BR-TR-2006-0002. 311th Performance Enhancement Directorate, Performance Enhancement Research Division, 2485 Gillingham Drive, Brooks City-Base, TX 78235-5105 9.: United States Air Force Performance Enhancement Research Division, Feb. 2006, p. 31. URL: http://www.dtic.mil/get-tr-doc/pdf?AD=ADA444925.

[19] D. Kuhl Mitchell. *Mental Workload and ARL Workload Modeling Tools*. Tech. rep. ARL-TN-161. Army Research Laboratory, Apr. 2000. URL: http://www.dtic.mil/dtic/tr/fulltext/u2/a377300.pdf.

[20] A. Hobbs and B. Lyall. *Human factors guidelines for remotely piloted aircraft system (RPAS) remote pilot stations (RPS)*. Tech. rep. National Aeronautics and Space Administration (NASA), July 2016. DOI: 10.13140/RG.2.2.12562.45768.

[21] Wayne Chappelle, Amber Salinas, and Kent McDonald. “Psychological Health Screening of Remotely Piloted Aircraft (RPA) Operators and Supporting Units”. In: (2011). URL: http://www.dtic.mil/dtic/tr/fulltext/u2/a582856.pdf.

[22] Walter W. Johnson et al. “NASA/CP-2013-216513 NASA’s Single -Pilot Operations Technical Interchange Meeting : Proceedings and Findings”. In: April (2013). URL: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140008907.pdf.

[23] Bureau of Transportation Statistics. *Air Carrier Financial Statistics*. 2015. URL: https://www.bts.gov/browse-statistical-products-and-data/bts-publications/air-carrier-financial-statistics-yellow-book.

[24] A. Schulte, D. Donath, and F. Honecker. "Human-System Interaction Analysis for Military Pilot Activity and Mental Workload Determination". In: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. Oct. 2015, pp. 1375–1380. DOI: 10.1109/SMC.2015.244.

[25] A. E. Ortiz and C. Langbort. "On multi-UAV scheduling for human operator target identification". In: June 2011, pp. 1837–1842. ISBN: 978-1-4577-0080-4. DOI: 10.1109/ACC.2011.5991585.

[26] A. D. Churchman et al. "Controls Automation and Task Allocation program". In: *Proceedings of National Aerospace and Electronics Conference (NAECON'94)*. May 1994, 587–597 vol.1. DOI: 10.1109/NAECON.1994.332852.

[27] Ting Chen. "Management of multiple heterogeneous unmanned aerial vehicules through transparency capability". Theses. Télécom Bretagne ; Université de Bretagne Occidentale, Feb. 2016. URL: https://hal.archives-ouvertes.fr/tel-01577924.

[28] European Commission. *Deployable Search and Rescue Integrated Chain with Unmanned systems Project (DARIUS)*. 2012. URL: http://www.spacetecpartners.eu/sites/default/files/attachments/08-day1-darius_project_overview.pdf; https://cordis.europa.eu/project/rcn/102362_en.html.

[29] International Civil Aviation Organization (ICAO). *Global Operational Data Link Document (GOLD) 2nd Ed.* 2013. URL: https://icao.int/APAC/Documents/edocs/GOLD_2Edition.pdf.

[30] Maastricht Upper Area Control Centre. *Don't text while driving. Text while flying!* 2018. URL: https://www.eurocontrol.int/sites/%20default/files/publication/files/factsheet-cpdlc.pdf.

[31] T. Lennertz, K. Cardosi, and A. Bisch. *Analysis of Controller-Pilot Voice Communications from Kansas City Air Route Traffic Control Center*. Tech. rep. DOT-VNTSC-FAA-17-13 ; DOT/FAA/TC-17/44. Federal Aviation Administration, July 2017. URL: https://rosap.ntl.bts.gov/view/dot/12501/dot_12501_DS1.pdf.

[32] Harris Corporation. *Federal Aviation Administration (FAA) Data Communications (Data Comm) User Information*. 2018. URL: https://www.harris.com/content/federal-aviation-administration-faa-data-communications-data-comm-user-information.

[33] Federal Aviation Administration. *NextGen Data Communications Program*. 2017. URL: https://www.faa.gov/nextgen/programs/datacomm.

[34] Eurocontrol. *Current Implementation Status*. 2017. URL: https://ext.eurocontrol.int/WikiLink/index.php/Current_Implementation_Status.

[35] M. Matyas. "Boeing Air Traffic Services (ATS) Data Link Perspectives". In: *DECEA CNS/ATM Workshop*. 2017. URL: http://clima.icea.gov.br/pesquisa/I-workshopcyberswim/downloads/E_WorkshopCyberSWIM_Boeing_ATS-Data-Link.pdf.

[36] International Civil Aviation Organization (ICAO). "Doc 9694-AN/955 Manual of Air Traffic Services Data Link Applications. First Edition". 1999. URL: https://icao.int/APAC/Documents/edocs/GOLD_2Edition.pdf.

[37] D. Comerford et al. “NASA’s Single-Pilot Operations Technical Interchange Meeting: Proceedings and Findings”. In: 2013. URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140008907.pdf.

[38] NAVAIRSYSCOM. *Flight Crew: Triton AirCrew*. 2016. URL: https://youtu.be/RqaSxanJJ80.

[39] The Guardian. *Boeing raises prospect of only one pilot in the cockpit of planes*. 2018. URL: https://www.theguardian.com/world/2018/feb/09/boeing-raises-prospect-of-only-one-pilot-in-the-cockpit-of-planes.

[40] Independent. *Single-pilot passenger planes could soon take to the skies, says Boeing*. 2018. URL: https://www.independent.co.uk/travel/news-and-advice/single-pilot-plane-boeing-autonomous-jet-technology-cockpit-a8506301.html.

[41] UAS Vision. *Only One in Eight Willing to Fly on Unmanned Aircraft*. 2018. URL: https://www.uasvision.com/2018/09/28/only-one-in-eight-willing-to-fly-on-unmanned-aircraft/.

[42] AviationWeek. *NASA Advances Single-Pilot Airliner Possibilities*. 2015. URL: https://youtu.be/rgdcdj5ietg.

[43] *NtoM ConOps Website*. URL: http://recerca.ac.upc.edu/ntom.

[44] Avionics International. *Control Six MQ-9 Reapers From Your Laptop With GA-ASI’s Heresy Software*. Oct. 2018. URL: https://www.aviationtoday.com/2018/10/19/gaasimmc/.

[45] C. A. Miller et al. “A Playbook Approach to Variable Autonomy Control: Application for Control of Multiple , Heterogeneous Unmanned Air Vehicles”. In: *American Helicopter Society 60th Annual Forum, Baltimore*. 2004. URL: https://pdfs.semanticscholar.org/7650/b5a006a156d6a46497d48c6d4def5e722b43.pdf.

[46] *Electronic Code of Federal Regulations*. 2018. URL: https://www.ecfr.gov.

[47] R. Jay Shively et al. “Human Performance Considerations for Remotely Piloted Aircraft Systems (RPAS)”. In: *Remotely Piloted Aircraft Systems Panel Second Meeting (RPASP/2), Montreal, Canada*. Mar. 2015.

[48] Eurocontrol. *System Wide Information Management (SWIM)*. 2018. URL: https://eur-registry.swim.aero/.

[49] Eurocontrol. *Network Operations Portal(NOP)*. 2018. URL: https://www.public.nm.eurocontrol.int/PUBPORTAL/gateway/spec/index.html.

[50] Eurocontrol. *OneSky’s DDR2*. 2018. URL: https://www.eurocontrol.int/articles/ddr2-web-portal.

[51] H. Wang, D. Gong, and R. Wen. “Air traffic controllers workload forecasting method based on neural network”. In: *The 27th Chinese Control and Decision Conference (2015 CCDC)*. May 2015, pp. 2460–2463. DOI: 10.1109/CCDC.2015.7162334.

[52] UAS Vision. *European MALE RPAS Programme Passes System Preliminary Design Review*. 2018. URL: https://www.uasvision.com/2018/12/17/european-male-rpas-programme-passes-system-preliminary-design-review/.

[53] C. Bierbaum, S. Szabo, and T. Aldrich. *Task analysis of the UH-60 mission and decision rules for developing a UH-60 workload prediction model*. Tech. rep. Technical Report No. AD-A210763. U.S. Army Res. Inst. Behav. Social Sci. Belvoir, VA, 1989.

[54] Stacey D. Scott et al. *Designing Decision and Collaboration Support Technology for Operators in Multi-UAV Operations Teams*. Tech. rep. Massachusetts Institute of Technology. Dept. of Aeronautics, Astronautics. Humans, and Automation Laboratory, 2007. URL: http://hdl.handle.net/1721.1/46729.

[55] S. S. Mulgund and G. L. Zacharias. "A Situation-Driven Adaptive pilot/Vehicle Interface". In: *Proceedings Third Annual Symposium on Human Interaction with Complex Systems (HICS'96)*. Vol. IEEE Computer Society Press. Aug. 1996, pp. 193–198. DOI: 10.1109/HUICS.1996.549515.

[56] Y. Brand and A. Schulte. "Model-based prediction of workload for adaptive associate systems". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Oct. 2017, pp. 1722–1727. DOI: 10.1109/SMC.2017.8122864.

[57] R. Little et al. *Crew reduction in armored vehicles ergonomic study (CRAVES)*. Tech. rep. ARL-CR-80. U.S. Army Research Laboratory, 1993. URL: http://www.dtic.mil/dtic/tr/fulltext/u2/a102719.pdf.

[58] A.T. Welford. "Single-channel operation in the brain". In: *Acta Psychologica* 27 (1967), pp. 5–22. ISSN: 0001-6918. DOI: https://doi.org/10.1016/0001-6918(67)90040-6.

[59] D. Kahneman. *Attention and effort*. Prentice Hall, 1973. ISBN: 978-0130505187. URL: https://www.princeton.edu/~kahneman/docs/%20attention_and_effort/Attention_hi_quality.pdf.

[60] B. L. Hooey et al. "The Underpinnings of Workload in Unmanned Vehicle Systems". In: *IEEE Transactions on Human-Machine Systems* 48.5 (Oct. 2018), pp. 452–467. ISSN: 2168-2291. DOI: 10.1109/THMS.2017.2759758.

[61] U.S. Army Research Laboratory Human Research & Engineering Directorate (ARL-HRED). *ARL IMPRINT*. Tech. rep. URL: https://www.arl.army.mil/www/default.cfm?page=3200.

[62] C. Wickens. *Processing Resources in Attention, Dual Task Performance, and Workload Assessment*. Tech. rep. EPL-81-3/ONR-81-3. University of Illinois Urbana-Champaign Engineering-Psychology Research Laboratory, July 1981, p. 59. URL: http://www.dtic.mil/dtic/tr/fulltext/u2/a102719.pdf.

[63] Christina F. Rusnock and Brett J. Borghetti. "Workload profiles: A continuous measure of mental workload". In: *International Journal of Industrial Ergonomics* 63 (Jan. 2018). Human Performance Modeling, pp. 49–64. ISSN: 0169-8141. DOI: https://doi.org/10.1016/j.ergon.2016.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0169814116301287.

[64] Dale Richards, Kurtulus Izzetoglu, and Graham Shelton-Rayner. "UAV Operator mental workload - A neurophysiological comparison of mental workload and vigilance". In: *AIAA Modeling and Simulation Technologies Conference*. June 2017. DOI: 10.2514/6.2017-3670.

[65] Norashiken Othman and Fairuz Romli. "Mental Workload Evaluation of Pilots Using Pupil Dilation". In: *International Review of Aerospace Engineering (IREASE)* 9 (June 2016), p. 80. DOI: 10.15866/irease.v9i3.9541.

[66] B.K. Siegel and K.J. Keller. "Pilot task monitoring using neural networks". In: June 1992, 709–714 vol.2. ISBN: 0-7803-0652-X. DOI: 10.1109/NAECON.1992.220517.

[67] S. G. Hart and C. D. Wickens. *Cognitive Workload*. Tech. rep. NASA/SP-2010-3407. 2010. URL: https://ston.jsc.nasa.gov/collections/TRS/_techrep/SP-2010-3407.pdf.

[68] R. Grier et al. "The Red-Line of Workload: Theory, Research, and Design". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 52.18 (2008), pp. 1204–1208. DOI: 10.1177/154193120805201811.

[69] P. U. Lee et al. "A Non-Linear Relationship between Controller Workload, Task Load, and Traffic Density: The Straw that Broke the Camel's Back". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 49. 12. 2005, pp. 1129–1133. DOI: 10.1177/154193120504901206.

[70] J. Stemberger, R. S. Allison, and T. Schnell. "Thermal Imaging as a Way to Classify Cognitive Workload". In: *2010 Canadian Conference on Computer and Robot Vision*. May 2010, pp. 231–238. DOI: 10.1109/CRV.2010.37.

[71] P. G. Raeth and J. M. Reising. "A Model of Pilot Trust & Dynamic Workload Allocation". In: *Proceedings of the IEEE 1997 National Aerospace and Electronics Conference. NAECON 1997*. Vol. 1. July 1997, 49–56 vol.1. DOI: 10.1109/NAECON.1997.617760.

[72] Gloria Calhoun et al. "Multiple Remotely Piloted Aircraft Control: Visualization and Control of Future Path". In: *VAMR 2013: Virtual, Augmented and Mixed Reality. Systems and Applications*. Ed. by Randall Shumaker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 231–240. ISBN: 978-3-642-39420-1. DOI: 10.1007/978-3-642-39420-1_25.

[73] M. R. Endsley and C. A. Bolstad. "Human Capabilities and Limitations in Situation Awareness". In: *Advisory Group for Aerospace Research and Development (AGARD) 520 Conference Proceedings*. Oct. 1992. URL: https://www.sto.nato.int/publications/AGARD/AGARD-CP-520/AGARDCP520.pdf.

[74] Eurocontrol. *Current Implementation Status*. 2017. URL: https://ext.eurocontrol.int/WikiLink/index.php/Current_Implementation_Status#Currently_Operational.

[75] Terence P. Fan and James Kuchar. "Evaluation of interfaces for pilot-air traffic control data link communications". In: *18th Digital Avionics Systems Conference*. Vol. 1/17 pp. vol.1. Dec. 1999, 4.A.5–1. ISBN: 0-7803-5749-3. DOI: 10.1109/DASC.1999.863723.

[76] Harris Corporation. *FAA's Controller Pilot Data Link Communications Services Instructional Video*. 2015. URL: https://www.youtube.com/watch?v=xUSFgkEyVAU.

[77] Eurocontrol Experimental Center (EEC). *Air/Ground Automated Tool*. URL: http://www.eurocontrol.int/sites/default/files/content/visuals/mini-sites/link2000/mcdu_rockwellcollins_1_1.html.

[78] K. L. Vu et al. "Influence of UAS pilot communication and execution delay on controller's acceptability ratings of UAS-ATC interactions". In: *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*. Oct. 2013. DOI: 10.1109/DASC.2013.6712624.

[79] Eurocontrol TV. *CPDLC request to climb*. 2012. URL: https://youtu.be/EK87g6Nlcyk.

[80] G. A. Shapiro. "Integrating Information from Controller to Pilot Data Link Communication (CPDLC) Messages". U.S. Patent No. US 9,911,337 B1. 2018.

[81] R. Kumar et al. "Systems and Methods for Context Based CPDLC". U.S. Patent No. US 2016/0035227 A1. 2016.

[82] Krishna et al. "Integrated Controller-Pilot Datalink Communication Systems and Methods for Operating the Same". U.S. Patent No. 10,026,322 B2. July 2018.

[83] Pakki et al. "System and Method for Providing Advisory Support Information on Downlink Clearance and Reports". U.S. Patent No. 9,224,301 B2. Dec. 2015.

[84] Letsu-Dake et al. "Systems and Methods for Improving an In-Trail Procedures Request". U.S. Patent No. 9,558,668 B2. Jan. 2017.

[85] UAS Vision. *Lost Satellite Link Caused Predator Loss in 2017*. 2019. URL: https://www.uasvision.com/2019/01/09/lost-satellite-link-caused-predator-loss-in-2017/.

[86] K. Cardosi and T. Lennertz. *Flight Deck Human Factors Issues for National Airspace System (NAS) En Route Controller Pilot Data Link Communications (CPDLC)*. Tech. rep. U.S. Department of Transportation Federal Aviation Administration Human Factors Division (ANG-C1), 2017.

[87] IFALPA Safety Bulletin. *Navigational errors on the North Atlantic*. Sept. 2011. URL: https://www.ifalpa.org/%20media/2214/12sab013-navigation-errors-on-the-north-atlantic.pdf.

[88] T. Lennertz and K. Cardosi. *Flightcrew Procedures for Controller Pilot Data Link Communications (CPDLC)*. Tech. rep. DOT/FAA/TC-15-12. Federal Aviation Administration, 2015.

[89] V. N. Nguyen and H. Holone. "N-best list re-ranking using syntactic score: A solution for improving speech recognition accuracy in air traffic control". In: *2016 16th International Conference on Control, Automation and Systems (ICCAS)*. Oct. 2016, pp. 1309–1314. DOI: 10.1109/ICCAS.2016.7832482.

[90] Lacko et al. "Aircraft Systems and Methods for Detecting Non-Compliant Pilot Action". U.S. Patent No. US 9,446, 852 B2. Sept. 2016.

[91] ICAO. *Remotely Piloted Aircraft System (RPAS) Concept of Operations for International IFR Operations*. Mar. 2017. URL: https://www.icao.int/safety/ua/documents/rpas%20conops.pdf.

[92] ICAO. *Annex 2 to the Convention on International Civil Aviation*. Tech. rep. July 2005.

[93] M. Perez-Batlle et al. "Real-time Simulations to Evaluate RPAS Contingencies in Shared Airspace". In: *5th SESAR Innovation Days - Book of abstracts*. 2015. URL: http://hdl.handle.net/2117/85618.

[94] K. D. Bilimoria, W. W. Johnson, and P. C. Schutte. "Conceptual Framework for Single Pilot Operations". In: *HCI-Aero '14 Proceedings of the International Conference on Human-Computer Interaction in Aerospace*. 4. Santa Clara, California, 2014. ISBN: 978-1-4503-2560-8. DOI: https://doi.org/10.1504/IJSTM.2003.003626.

[95] Eurocontrol. *Early Demonstration & Evaluation Platform Project (eDEP)*. 2002. URL: https://www.eurocontrol.int/eec/public/standard_page/ERS_edep.html.

[96] RTI. *Connext DDS, Traninig and Simulation*. URL: https://www.rti.com/industries/aerospace-defense/training-simulation.

[97] J. S. McCarley and C. D. Wickens. *Human Factors Implications of UAVs in the National Airspace*. Tech. rep. Technical Report AHFD-05–05/FAA-05–01. Federal Aviation Administration (FAA), 2005.

[98] Eurocontrol. *What is the Flight Object?* URL: https://www.eurocontrol.int/services/flight.

[99] E. Pastor et al. "In-Flight Contingency Management for Unmanned Aerial Vehicles". In: *Journal of Aerospace Computing, Information, and Communication* 9.4 (2012). DOI: 10.2514/1.55109.

[100] SKYbrary. *CPDLC (SKYclip)*. Oct. 2018. URL: https://www.skybrary.aero/index.php/CPDLC_(SKYclip)?utm_source=SKYbrary&utm_campaign=31f0d148be-612_CPDLC_SKYclip_17_12_2018&utm_medium=email&utm_term=0_e405169b04-31f0d148be-276597409.

[101] IFALPA. *Potential CPDLC Message Delivery Latency*. Dec. 2017. URL: https://www.ifalpa.org/media/%202190/17sab12-potential-cpdlc-message-delivery-latency.pdf.

[102] European Organisation for Civil Aviation Equipment (EUROCAE). *Safety and Performance Standard for Baseline 2 ATS Data Communications (Baseline 2 SPR Standard)*. 2014.

[103] Eurocontrol. *ATC Data Link Operational Guidance in support of DLS Regulation*. 2012. URL: https://www.eurocontrol.int/sites/default/files/article/content/documents/nm/link2000/atc_dl_oper_guidance_for_link2000_services_v6_0.pdf.

[104] ICAO. *Global Flight Information Region*. 2008. URL: https://gis.icao.int/icaofir/.

[105] Belgocontrol. *Aeronautical Information Publication for Belgium (section GEN-3.4)*. Oct. 2018. URL: https://www.belgocontrol.be/html/belgocontrol_static/eaip/eAIP_Main/html/eAIP/EB-GEN-3.4-en-GB.html.

[106] T. D. Judd et al. "Enhanced Flight Crew Display for Supporting Multiple Controller/Pilot Data Link Communications (CPDLC) Versions". U.S. Patent No. 9,626,872 B2. July 2017.

[107] Dominic et al. "System and Method of Integrating Data Link Messages with a Flight Plan". U.S. Patent No. 9,881,504 B2. 2018.

[108] Johnson et al. "System and Method for Providing a Real Time Audible Message to a Pilot". U.S. Patent No. 9,371,140 B1. 2016.

[109] Claudiu-Mihai Geacar. "Reducing Pilot/ATC Communication Errors Using Voice Recognition". In: *27th International Congress of the Aeronautical Sciences*. 2010. URL: http://www.icas.org/ICAS_ARCHIVE/ICAS2010/PAPERS/441.PDF.

[110] Eurocontrol Experimental Centre. *Technical Analysis of ATC Controller to Pilot Voice Communication with Regard to Automatic Speech Recognition Systems*. Tech. rep. EEC Note No. 01/2001. Eurocontrol, Jan. 2001. URL: https://www.eurocontrol.int/eec/gallery/content/public/document/eec/report/2001/001_Analysis_of_Controller_Pilot_Voice_Communication.pdf.

[111] A. Lechner, P. Mattson, and K. Ecker. "Voice recognition: software solutions in real-time ATC workstations". In: *IEEE Aerospace and Electronic Systems Magazine* 17.11 (Nov. 2002), pp. 11–16. ISSN: 0885-8985. DOI: 10.1109/MAES.2002.1047373.

[112] O. V. Prinzo. "How data link communication might affect controller workload on a terminal option". In: *17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267)*. Vol. 2. Oct. 1998, F51/1–F51/8 vol.2. DOI: 10.1109/DASC.1998.739830.

[113] C.M. Geacar. "Reducing Pilot/Atc Communication Errors Using Voice Recognition". In: *27th International Congress of the Aeronautial Sciences*. 2010.

[114] Eurocontrol. *Technical Analysis of ATC Controller to Pilot Voice Communication with Regard to Automatic Speech Recognition Systems*. Tech. rep. EEC Technical/Scientific Report No. 2001-001. Eurocontrol Experimental Centre, 2001. URL: https://www.eurocontrol.int/eec/public/standard_page/DOC_Report_2001_001.html.

[115] M.L. Cummings. "The Need for Command and Control Instant Message Adaptive Interfaces: Lessons Learned from Tactical Tomahawk Human-in-the-Loop Simulations". In: *CyberPsychology & Behavior* 7.6 (2004). PMID: 15687799, pp. 653–661. DOI: 10.1089/cpb.2004.7.653.

[116] C. D. Wickens. "Multiple resources and performance prediction". In: *Theoretical Issues in Ergonomics Science* 3.2 (2010), pp. 159–177. DOI: 10.1080/14639220210123806.

[117] K. M. Cardosi. “Time Required for Transmission of Time-Critical Air Traffic Control Messages in an En Route Environment”. In: *The International Journal of Aviation Psychology* 3.4 (1993), pp. 303–313. DOI: 10.1207/s15327108ijap0304_4.

[118] Axel Schulte and Diana Donath. “Measuring Self-adaptive UAV Operators’ Load-Shedding Strategies under High Workload”. In: *Engineering Psychology and Cognitive Ergonomics*. Ed. by Don Harris. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 342–351. ISBN: 978-3-642-21741-8. DOI: https://doi.org/10.1007/978-3-642-21741-8_37.

[119] J. C. Sperandio. “Variation of Operator’s Strategies and Regulating Effects on Workload”. In: *Ergonomics* 14.5 (1971). PMID: 5148233, pp. 571–577. DOI: 10.1080/00140137108931277.

[120] S.G. Hart. “Crew workload-management strategies - A critical factor in system performance”. In: *Fifth International Symposium on Aviation Psychology, Columbus, OH*. 19900039124. Jan. 1989. URL: https://ntrs.nasa.gov/search.jsp?R=19900039124.

[121] Dario Salvucci and Peter Bogunovich. “Multitasking and monotasking: The effects of mental workload on deferred task interruptions”. In: vol. 1. Jan. 2010, pp. 85–88. DOI: 10.1145/1753326.1753340.

[122] Christopher D. Wickens, Diane L. Sandry, and Michael Vidulich. “Compatibility and Resource Competition between Modalities of Input, Central Processing, and Output”. In: *Human Factors* 25.2 (1983). PMID: 6862451, pp. 227–248. DOI: 10.1177/001872088302500209.

[123] Christopher D. Wickens, Stephen R. Dixon, and Dervon Chang. *Using Interference Models to Predict Performance in a Multiple-Task UAV Environment – 2 UAVs*. Tech. rep. Technical report for Micro Analysis and Design, 2003.

[124] Mark Draper et al. “Manual Versus Speech Input for Unmanned Aerial Vehicle Control Station Operations”. In: vol. 47. 1. 2003, pp. 109–113. DOI: 10.1177/154193120304700123.

[125] Federal Aviation Administration (FAA). *Operational Authorization Process for Use of Data Link Communication System*. 2015.

[126] H. Glaser-Opitz, L. Glaser-Opitz, and J. Labun. “Data Link Communication Interface with Flight Simulator in Form of a CPDLC”. In: *Proceedings of the International Scientific Conference Modern Safety Technologies in Transportation*. 2015.

[127] IBM. *CPLEX Optimizer*. URL: https://www.ibm.com/analytics/cplex-optimizer.

ACKNOWLEDGEMENTS

The author would like to thank Javier Negre, Joan Pere Sánchez and Laura López for their collaboration in the experiments and to Marc Pérez, Dr. Alan Hobbs and Jerónimo Vicente for their kind provision of expert advice in aeronautics and in the RPAS area. Controllers behind the Twitter account @controladores provided their knowledge, experience and opinions about the CPDLC display and some of the measures suggested.

Real-Time Innovations (RTI) and its Lead Field Application Engineer Sara Granados, kindly provided us with the software license and the support to use their solution in the prototype.

Last but not least, my thanks to Pablo Royo and Cristina Barrado for being my cicerones during the doctorate.

FUNDING OF THE THESIS

This work was part of a thesis funded by the grant with reference BES-2014-071096 belonging to the Research Personnel Training sub-programme from the Ministry of Economy and Competitiveness of Spain, and partially by the Ministry of Economy and Enterprise of Spain under contract TRA2016-77012-R.