

Controller-Pilot Data Link Communications Display Oriented to Multiple Remotely Piloted Aircraft Pilots

Miguel-Ángel Fas-Millán, Enric Pastor

Abstract – A Controller-Pilot Data Link Communications (CPDLC) interface oriented to Remotely Piloted Aircraft Systems (RPAS) pilots was implemented as part of a synthetic task environment aimed to test the NtoM concept of operations (ConOps). This ConOps was envisioned to support multi-RPAS piloting in non-segregated airspace. Considering a long-term implementation, it assumes a future widespread use of CPDLC, exploiting its full potential to try to reduce the delay in the communication flow associated with unmanned aircraft and any possible added delay that concurrent piloting could add. Designed to be quick and intuitive, the current prototype of this display, on its own, could be used by pilots - of manned or unmanned aircraft - and controllers to practise and get used to the CPDLC message set, composition rules and procedures. Developed using the Data Distribution Service (DDS) standard, it allows the definition of different Quality of Service (QoS) scenarios for data communications, which can be used to train the procedures established for problems arising from faults in communications.

Copyright © 2019 The Authors.

Published by Praise Worthy Prize S.r.l. This article is open access published under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Keywords: RPAS, CPDLC, UAS, Data Link

Nomenclature

ATC	Air Traffic Control
ATM	Air Traffic Management
ATSU	Air Traffic Service Unit
CDA	Current Data Authority
CDU	Control Display Unit
ConOps	Concept of operations
CPDLC	Controller-Pilot Data Link Communications
DM	Downlink Message
FMS	Flight Management System
GCS	Ground Control Station
ICAO	International Civil Aviation Organization
MCDU	Multi-Function Control and Display Unit
NtoM	ConOps to which this display belongs
QoS	Quality of Service
RPAS	Remotely Piloted Aircraft System
UM	Uplink Message
W/U	WILCO/UNABLE

I. Introduction

Controller-Pilot Data Link Communications (CPDLC) implies using text messages instead of the voice radio channel. Its use has many benefits for all parties [1, Chapter 2.1.1] [2], like the increase of the sector capacity, decongestion of voice frequencies, avoidance of input errors by the pilot (some Air Traffic Controller (ATC) instructions can be directly loaded into the Flight Management System (FMS) [1, Chapter 5.3.5]),

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.

There are 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) [3] 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, or 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 readback and hearback errors; therefore, fewer NORDO (No Radio) problems and less frequency occupancy time - 20% of the time in the [3] analysis was spent on this process. CPDLC is just one application of the data link technology; [4] describes all data link applications and

services. 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 U.S. [5] and only for the Departure Clearance Service (DCL), on VDL Mode 0/A, with a first small set of other services envisioned to start use in a few months (NextGen's Data Comm program roadmap can be found in [6]). 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 Single European Sky concept and has been operational since 2003. This CPDLC covers a wider variety of uses, but has not been homogeneously implemented yet. [7] 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 [8]. The European CDPLC is based on the ATN/OSI protocol. Being a different protocol, both the European and North Atlantic CPDLC systems are incompatible.

NtoM [9] is a suggestion of concept of operations (ConOps) that aims to improve the awareness, productivity and safety of Remotely Piloted Aircraft System (RPAS) pilots concurrently piloting several aircraft in non-segregated airspace. One of the main challenges to make such operations feasible is avoiding any impact on the work of the Air Traffic Controllers (ATC) due to delays, oversights or errors caused by that parallel piloting. But CPDLC provides a great basis to build tools addressed at avoiding those problems. Considering what could only be a long-term implementation of the ConOps, when the widespread use of CPDLC should be a fact, it was assumed as the main channel of communication and the safety measures implemented in the NtoM prototype leverage its potential. Pilots required, of course, a CPDLC display and, considering that a Ground Control Station (GCS) does not suffer from the constraints of a cockpit, it was an opportunity to develop a display with a better usability than the current communications display units (CDU), which would reduce the head-down. One problem arising during the multi-RPAS control could be an excessive added increase of the response time to ATC, which would not be operationally viable. Current voice communications of unmanned aircraft add a delay that is mainly accepted by controllers when it is short [10]. But at the moment of writing this document, no comparison

was found regarding delay times of RPAS pilots vs. manned aircraft pilots when using CPDLC. A drawback of CPDLC with respect to voice communications is the time required to compose and send the message in comparison with voice communications. 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) [11]. Taking a look at these interfaces and how they work, with a menu navigation very similar to the 1990s Automatic Teller Machines, some of them even monochromatic (find a video example in [12] or a simulator in [13]), it is easy to guess that even the less ambitious current computer graphic user interface can easily improve their usability. Such an improvement would help to reduce the head-down as required in [4]. If the communications of a RPAS pilot can suffer some delay due to the latency of the data link, and if such a response time can be increased because the pilot is attending another flight, then reducing as much as possible the time required to use the display was key to not ballasting the work of the controllers in the multi-RPAS scenario even more. In the NtoM prototype, the CPDLC displays in the controller and pilot clients are almost identical, although less effort was put into the ATC version, as it was only required to interact with the pilots while performing the experiments, which were focused on testing the safety measures implemented for the pilot interface. Current ATC tools allow messages to be composed and sent by opening a dialogue from the same flight radar tag (see this in action in [14]). A similar approach has been suggested for the manned aircraft [15], 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 tool. This option seems to be sufficient for simpler messages, and it has the benefit that the dialogue is so minimised that pilots do not need to move their sight to a side panel or monitor to compose the message.

Maybe the display suggested here, which is more intuitive, user friendly and informative, would 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 appropriate for a pilot. Although [15] offers the possibility to navigate through the message history, this is done as in a common MCDU, with a high head-down.

But one virtue of the [15] flight deck situational

awareness communication system 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. As will be explained in the description of the features of the display suggested here, it coincides with [15] in the suitability of a button to execute the instruction and the automated reply when used. Another suggested approach [16] 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 addition 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 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 [17], 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 glareshield, 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 [18], 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, [19] allows a quick composition of the messages to request an In-Trail Procedures (ITP) altitude change. In any case, for the time being, the software presented here could be used by controllers and pilots to practise the CPDLC procedures. As reported in [20], 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) [21, 8.3].

The NtoM CPDLC display allows scripts to be run simulating the role of the ATC - as 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 RTI Connex connectivity framework [22], compliant with the Data Distribution Service (DDS) standard, which allows the definition of different scenarios of Quality of Service (QoS). In this way, users can face different levels of communication reliability and train in the procedures established for the problems arising from it. In any event, the present display does not try to mimic the current cockpit displays as other training tools already do [23]; then, ultimately, manned aircraft pilots should also practise the menu navigation and behaviour of the specific unit implementation to be used. Although NtoM implements several measures at different levels to minimise the impact of multitasking, this document focuses only on those decisions regarding communications. Despite the CPDLC procedures described here following standard recommendations [24], [25], 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 that they were completely correct and resolve some doubts about the interpretation of the documentation. A subject matter expert review would be necessary prior to using this software as an educational tool. First, the document details the parts of the interface, which information is shown and how it is displayed. Then it describes the message composition and the help provided to avoid errors and ambiguity. Finally, the outlined voice recognition option is presented. Throughout the document, to reduce as much as possible the number of screenshots and to clearly show how the application works, links to screencasts are provided.

II. Display Description and Behaviour

Fig. 1 shows both the pilot (left) and ATC displays.

The main difference that can be noted is that they have a different message element set and hotkeys available.

This section describes the parts, operation, and the reasons behind the design decisions.

II.1. Message History

In order 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, having the log of messages within view helps to recall at a glance the conversation, with no need to navigate to load previous messages. 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.

The reason is that more than one dialogue could be open at the same time, related to different communications; the colour helps to distinguish them. Within each dialogue, messages sent are coloured with a lower saturation to highlight the received ones. See Fig. 1 or [26] 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 problems identified when using them. 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. The solution of some display manufacturers is to disable the buttons to reply until the message is navigated and displayed to its end. In this respect, the procedure recommended by some airlines to their pilots for the long messages is to print them, a practice which is

not recommended as printers are not always reliable [27].

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. The messages in the history print the variables of the element in bright blue, to make the key part of the text stand out (Fig. 1). Each message element is printed in a different row, and the whole message shows its timestamp. The timestamp is important to identify messages with 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 [28]. The alert level, information of interest mainly for the controllers, who can use it to decide which message to attend to first, is printed in a different colour depending on the urgency.

Accordingly, when a message is received, the aural warning is different depending on its alert level. When possible, an Apply button can be found next to the received message in the pilot display. This would send to the aircraft the commands required to execute the orders in the message. To allow this, a GCS providing an interface to bi-directionally communicate with the display is required. For the present prototype, which used the eDEP [29] simulator, a basic interface was coded on it to receive the commands of the display, and the limitations of the simulator only allow a few instructions like changes of speed, level, heading and direct to waypoint to be loaded.

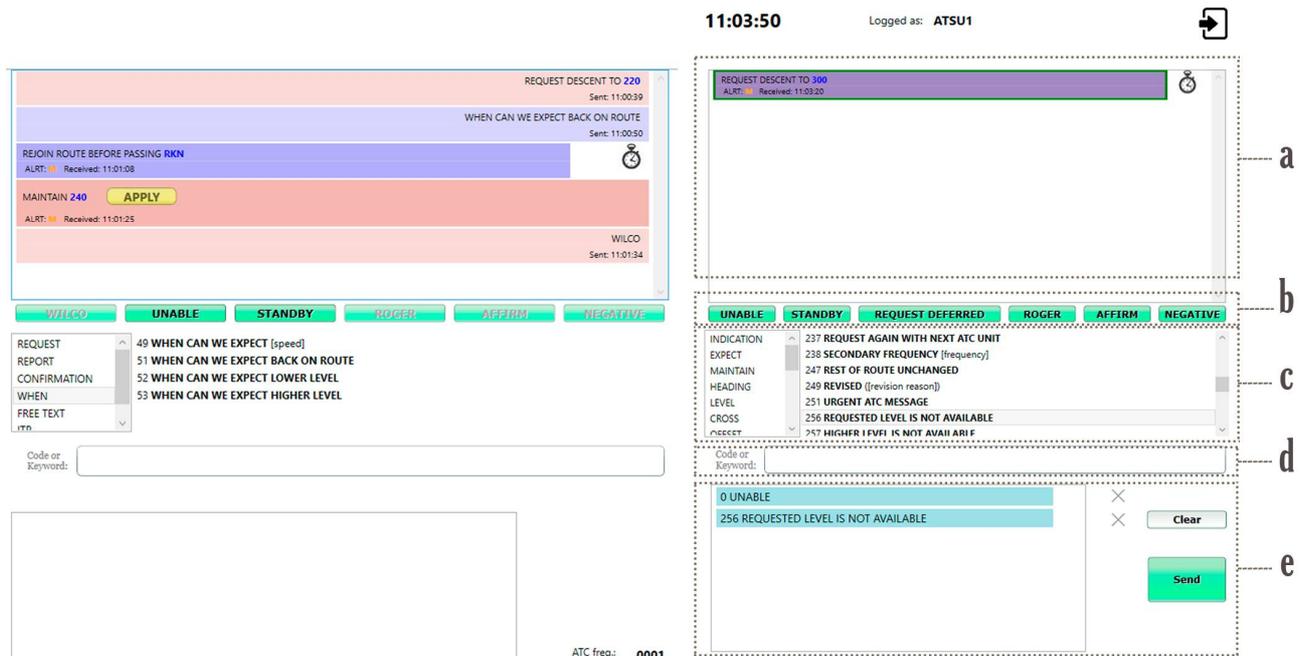


Fig. 1. Pilot (left) and controller CPDLC displays (the situations depicted do not belong to the same communication). Parts: a) message history; b) hotkeys; c) message element selection by category; d) element selection by code or keyword; e) message composition

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 [25], 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 [24, 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 to the ATC; for instance, in a message with a unique element UM20 CLIMB TO REACH [*level*]. But 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 WILCO/UNABLE (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 to 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*]); or b) sending a first message with a single element for the element with highest precedence (the W/U), and then sending a separate message with the answer to the other element(s).

Choosing 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 will not be there to claim for the second message. In order 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 would issue a warning if any of the elements does not have its required reply. A screencast illustrates this in [30]. Open messages are those waiting for an answer. These show a chronometer icon 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, briefly postpone the reply to attend another flight, and then simply forget to send it. The solution chosen was to play a synthetic voice warning that there were 30 s left for the timeout.

II.2. Message Composition

The statement in [4, ch. 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 has at least one 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. 2). 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 (Fig. 1, c) appearing at the left side of this block. These are determined based on the description of the elements provided in the documentation, although users should be surveyed about other categorisations preferred. 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.



Fig. 2. Hotkeys availability associated to the allowed replies of the message selected

An idea that is 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. The second selection option is the input box labelled Code or Keyword (Fig. 1, 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 [31].

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. 3). 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 is 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” [25], and a text box forces the pilot to write the unit abbreviation, which is slower than selecting it from a dropdown. Regarding numerical values, a selector makes sense when the range and step do not result in excessive scrolling to find the value.

Once the values have been filled in, the Add button includes the element in the message (Fig. 1, e). 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.

II.3. 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. These hotkeys remain active only when they make sense considering the selected message to be answered. In Fig. 2 the pilot receives a message that, considering the reply precedence of the elements, should be only replied with a 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, as long as there was space available.

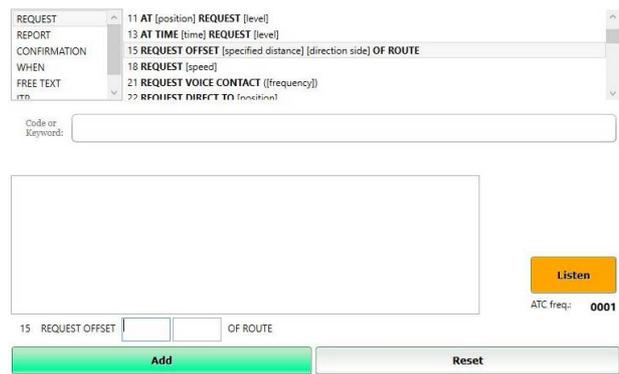


Fig. 3. Each element variable has its own input box, subject to its associated value error checks

II.4. Guiding 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, just some illustrative examples were included in the current prototype. Some can be seen in [14]. A feature that is envisioned, as part of the guiding, is 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 will be available during the gradual implementation of the service, and it can change among ATSU's 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 is scheduled to start in 2019, and 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. The detailed list of elements allowed appears in [27]. Europe uses an element message set defined in the Link2000+ Programme [32], which is a subset of the ICAO's CPDLC [1]; on it, some elements are mandatory to support by the European ATS authorities providing CPDLC, and others are optional. For instance, the Maastricht Upper Area Control Centre handles a wider message element set - detailed in [33] - than the one envisioned in the U.S., but free text messages are not allowed neither. 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 limit the input units, which are also affected by the version. The appropriate behaviour would be updated with each change of Current Data Authority (CDA), which is the ATSU communicating with the aircraft. 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 [34] 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 feature suggested in the present article would also take into account the ATSU-specific optional elements and warnings about specific uses in the area, which is allowed by keeping a database of CPDLC service availability at an ATSU level, instead of at a version level, which would be a better fit to the real supported services.

II.5. Voice-Controlled Composition

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 backup of

auditory information is useful when the user is busy attending to a task and cannot move sight to another source of information, particularly when the access to the information is slow. An illustrative example is the proposal [35] 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 that was partially developed at the moment of writing this document uses voice recognition. In NtoM, a pilot could be piloting several RPAS and an ATC message reaches an aircraft while the pilot is busy with another flight. The pilot could decide to wait a short time to answer the message, adding extra stress to the current operation and a delay in the response to the ATC. The interface offers the possibility to avoid such disturbances by verbally composing and sending the reply. First, when the message arrives for the flight in the background (a flight that is not the one currently selected), a synthesised voice reads the message aloud, identifying the receiver by its callsign. If the pilot does not want to verbally compose a reply right then, an "ok" will dismiss the voice assistant.

Otherwise, the pilot 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 pattern of the elements allow the engine to determine their syntax. 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 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 [36]. An operative implementation would require the advice and 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 output 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 [37], which clearly illustrates this challenge when using voice recognition to allow ATC to compose CPDLC messages. As mentioned in that work, from the controller's side, using voice 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 [38]. 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. [39] has also advocated for ATC CPDLC verbal composition as a tool to reduce communication errors, and [40] 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.

II.6. Automated Reports

This feature is part of the NtoM ConOps but it could be embedded in a standalone implementation of the display as long as it could communicate with the GCS to get the values requested in the report. Automatic Dependent Surveillance – Contract (ADS-C) is a data link application that 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 those, the ATC 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 an 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 kind of report could be automatically replied to in order to release the pilot from the task. This is the case with elements UM35, UM36 and UM37 (CONFIRM ASSIGNED [LEVEL/SPEED/ROUTE]), or REPORT ENDURANCE AND PERSONS ON BOARD. The ones requiring a condition to be met before being sent, like UM130R, would need monitoring to determine the moment. The screencast [41] 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 [42]. 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 to the waypoints with those scheduled actions. Those communication requirements could have been added during the flight also, 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 approach releases the pilot from the confirmation step and automatically replies if it is able to access the data required. With regard to associating communication actions to waypoints in the flight plan, the NtoM system uses a more general concept of tasks (e.g. a scheduled frequency change) that could

include them as any other planned task. While the required monitoring is placed on the server in the NtoM system, it could be embedded in a standalone version of the display as long as it had access to the aircraft readings.

III. Observations During the Experiments

While the experiments were addressed to evaluating the measures suggested by the ConOps, and not focused on the CPDLC display, it is worth mentioning some aspects observed during the interaction of the participants with it. They were not professional pilots, but all university graduates with long experience using computer interfaces. The scenarios simulated very high - and unrealistic - levels of workload, with the aim being to make the needs and weaknesses of the system arise when it comes to helping the pilot in overwhelming and unavoidable situations. As 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 to 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. The following observations are related 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 the need to place an open message icon in the flight list, 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 much 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). In order 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 of 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 (although, when possible, the system monitors and warns if they are delayed in 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. These observations seem to satisfy and get the benefits of Endsley's recommendations [43] 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 NtoM system in other parts of the pilot client, so will not be discussed here.

IV. Conclusion

NtoM is a ConOps addressed to pilots controlling several RPAs in non-segregated airspace. It assumes a future widespread implementation and the full use of the potential of CPDLC. In that future context, the data link should be the main channel of communication, and ATC and pilots would be trained to leverage the whole message element set and the possibility of greater complexity of the dialogue exchange. But current cockpit displays and dialogues embedded in GCSs or ATC tools are not prepared for this scenario, or involve a high head-down. The present display was conceived for the future of CPDLC, but also to minimise the head-down, considering that it was part of a ConOps of multi-RPAS pilots, which could add some delay in the response time if the pilot is attending some other flight when the message arrives.

The display presents the dialogue like an intuitive chat, and allows a quick selection of the preformatted message elements. Warnings and guidelines help users while learning the constraints and best practices of the message composition. Visual and aural warnings try to keep pilots aware of the status of the dialogue, to avoid timeouts. Some replies are automated when possible to release the pilot from the task and avoid typing errors. The usability measures suggested here aim to inspire future CPDLC displays, while the current prototype could serve as the basis for the training of Standard Operating Procedures (SOPs) for CPDLC usage by pilots and controllers. 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.

The use of DDS as communication middleware allows the definition of QoS profiles to simulate the possible problems arising from high latency or packet loss; this allows training to be given in the associated procedures. Before using it as an educational tool, this prototype should be first reviewed by subject matter experts to check that the documentation has been correctly interpreted and to determine the implementation of some specific issues.

Acknowledgements

This work is 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.

References

- [1] *Global Operational Data Link Document (GOLD)*, ICAO. 2013. https://icao.int/APAC/Documents/edocs/GOLD_2Edition.pdf.
- [2] Maastricht Upper Area Control Centre, *Don't text while driving. Text while flying!* 2018, <https://www.eurocontrol.int/sites/default/files/publication/files/factsheet-cpdlc.pdf>.
- [3] T. Lennertz et al., *Analysis of Controller-Pilot Voice Communications from Kansas City Air Route Traffic Control Center*, report prepared for the Federal Aviation Administration (FAA), 2017.
- [4] ICAO. Doc 9694-AN/955 *Manual of Air Traffic Services Data Link Applications*, 1999.
- [5] Harris Corporation, Federal Aviation Administration (FAA) *Data Communications (Data Comm) User Information*, 2018, <https://www.harris.com/content/federal-aviation-administration-faa-data-communications-data-comm-user-information>.
- [6] NextGen Data Communications Program <https://www.faa.gov/nextgen/programs/datacomm/>.
- [7] Eurocontrol, *Current Implementation Status*, 2017, https://ext.eurocontrol.int/WikiLink/index.php/Current_Implementation_Status.
- [8] M. Matyas, *Boeing Air Traffic Services (ATS) Data Link Perspectives*, DECEA CNS/ATM Workshop, 2017, http://clima.icea.gov.br/pesquisa/I-workshopcyberswim/downloads/E_WorkshopCyberSWIM_Boeing_ATS-Data-Link.pdf.
- [9] M.-A. Fas-Millán et al., NtoM: a Concept of Operations for Pilots of Multiple Remotely Piloted Aircraft, (2019) *International Review of Aerospace Engineering (IREASE)*, accepted for publication.
- [10] K. L. Vu, G. Morales, D. Chiappe, T. Z. Strybel, V. Battiste, J. Shively, T. J. Buker. Influence of UAS Pilot Communication and Execution Delay on Controller's Acceptability Ratings of UAS-ATC Interactions, *Digital Avionics Systems Conference (DASC), IEEE/AIAA 3rd*, 2013.
- [11] T. P. Fan, J. K. Kuchar. Evaluation of Interfaces for Pilot-Air Traffic Control Data Link Communications, *Digital Avionics Systems Conference*, 1999. Proceedings. 18th (1999). doi: 10.1109/DASC.1999.863723.
- [12] Harris Corporation, FAA's *Controller Pilot Data Link Communications Services Instructional Video*, 2015. <https://www.youtube.com/watch?v=xUSFgkEyVAU>.
- [13] Eurocontrol Experimental Center, *Air/Ground Automated Tool*, https://www.eurocontrol.int/sites/default/files/content/visuals/minisites/link2000/mcdu_rockwellcollins_1_1.html.
- [14] Eurocontrol TV, *CPDLC request to climb*, 2012, <https://youtu.be/EK87g6Nlcyk>.
- [15] G. A. Shapiro, *Integrating Information from Controller to Pilot Data Link Communication (CPDLC) Messages*, U.S. Patent No. US 9,911,337 B1, 2018.
- [16] R. Kumar et al., *Systems and Methods for Context Based CPDLC*, U.S. Patent No. US 2016/0035227 A1, issued 2016.
- [17] Krishna et al., *Integrated Controller-Pilot Datalink Communication Systems and Methods for Operating the Same*, U.S. Patent No.: 10,026,322 B2, issued Jul. 17, 2018.
- [18] Pakki et al., *System and Method for Providing Advisory Support Information on Downlink Clearance and Reports*, U. S. Patent No. US 9,224,301 B2, issued Dec. 29, 2015.
- [19] Letsu-Dake et al., *Systems and Methods for Improving an In-Trail Procedures Request*, U. S. Patent No. US 9,558,668 B2, issued Jan. 31, 2017.
- [20] T. Lennertz and K. Cardosi, *Flightcrew Procedures for Controller Pilot Data Link Communications (CPDLC)*, report prepared for the Federal Aviation Administration, DOT/FAA/TC-15-12, 2015.
- [21] Federal Aviation Administration (FAA), *Operational Authorization Process for Use of Data Link Communication System*, *Advisory Circular No. 120-70C*, 2015.
- [22] RTI Connex DDS, *Training and Simulation*, <https://www.rti.com/industries/aerospace-defense/training-simulation>.
- [23] H. Glaser-Opitz, L. Glaser-Opitz and J. Labun, *Data Link Communication Interface with Flight Simulator in Form of a CPDLC*, *Proceedings of the International Scientific Conference Modern Safety Technologies in Transportation*, 2015.
- [24] International Civil Aviation Organization (2013), *Global Operational Data Link Document (GOLD)* 2nd Ed., 2013. https://icao.int/APAC/Documents/edocs/GOLD_2Edition.pdf.
- [25] The European Organisation for Civil Aviation Equipment (EUROCAE), *Safety and Performance Standard for Baseline 2 ATS Data Communications* (Baseline 2 SPR Standard), 2014.
- [26] Dialogue interleaving screencast, http://recerca.ac.upc.edu/ntom/videos/NtoM_CPDLCDisplay_dialogue_interleaving.mp4.
- [27] K. Cardosi, T. Lennertz, *Flight Deck Human Factors Issues for National Airspace System (NAS) En Route Controller Pilot Data Link Communications (CPDLC)*, prepared for US Department of Transportation Federal Aviation Administration Human Factors Division (ANG-C1), 2017.
- [28] IFALPA, *Potential CPDLC Message Delivery Latency*, *IFALPA Safety Bulletin 8 Dec 2017* <https://www.ifalpa.org/media/2190/17sab12-potential-cpdlc-message-delivery-latency.pdf>.
- [29] Early Demonstration & Evaluation Platform (EDEP), <http://www.eurocontrol.fr/projects/edep/>.
- [30] Automated loading of instructions and reply screencast, http://recerca.ac.upc.edu/ntom/videos/NtoM_autoWilco.mp4.
- [31] Message composition screencast, http://recerca.ac.upc.edu/ntom/videos/NtoM_message_composition.mp4.
- [32] Link2000+, *ATC Data Link Operational Guidance in support of DLS Regulation No 29/2009*, Eurocontrol, 2012, https://www.eurocontrol.int/sites/default/files/article/content/documents/nm/link2000/atc_dl_oper_guidance_for_link2000_services_v6_0.pdf.
- [33] Belgocontrol, *Aeronautical Information Publication for Belgium* (section GEN-3.4) valid from 11 Oct 2018. https://www.belgocontrol.be/html/belgocontrol_static/eaip/eAIP_Main/html/eAIP/EB-GEN-3.4-en-GB.html.
- [34] T. D. Judd, M. J. Kayser, T. F. McGuffin and R. Gupta, *Enhanced Flight Crew Display for Supporting Multiple Controller/Pilot Data Link Communications (CPDLC) Versions*, U. S. Patent No. US 9,626,872 B2, 2017.
- [35] Johnson et al., *System and Method for Providing a Real Time Audible Message to a Pilot*, U. S. Patent No. US 9,371,140 B1 issued 2016.
- [36] http://recerca.ac.upc.edu/ntom/videos/NtoM_verbal_composition.mp4.
- [37] A. Lechner, P. Mattson, K. Ecker. Voice Recognition: Software Solutions in Real-time ATC Workstations. *IEEE Aerospace and Electronic Systems Magazine* (Volume: 17, Issue: 11, Nov. 2002) doi: 10.1109/MAES.2002.1047373.
- [38] O. V. Prinzo. How Data Link Communication Might Affect Controller Workload On a Terminal Option. *Digital Avionics Systems Conference (DASC), IEEE/AIAA 17th* (1998).
- [39] C. M. Geacar. Reducing Pilot/ATC Communication Errors Using Voice Recognition. *27th International Congress of the Aeronautical Sciences* (2010).
- [40] *Technical Analysis of ATC Controller to Pilot Voice Communication with Regard to Automatic Speech Recognition Systems*, (Eurocontrol Experimental Centre, 2001).
- [41] Automated reports screencast, http://recerca.ac.upc.edu/ntom/videos/NtoM_automated_reports.mp4.
- [42] Dominic et al., *System and Method of Integrating Data Link Messages with a Flight Plan*, U.S. Patent No. US 9,881,504 B2, issued in 2018.
- [43] M. R. Endsley and C. A. Bolstad, *Human Capabilities and Limitations in Situation Awareness*, *Advisory Group for Aerospace Research and Development (AGARD) 520 Conference Proceedings*, Edinburgh, 1992, <https://www.sto.nato.int/publications/AGARD/AGARD-CP-520/AGARDCP520.pdf>.

Authors' information

Polytechnic University of Catalonia.



Miguel-Ángel Fas-Millán is a computer engineer from the Universitat Jaume I (UJI) of Castelló, master in Computer Vision and Artificial Intelligence from the Universitat Autònoma de Barcelona (UAB), and currently Computer Architecture Ph.D. candidate at the Polytechnic University of Catalonia (UPC). The topic of the thesis is the development of a concept of operations for the integration of remotely piloted air systems in non-segregated airspace.



Enric Pastor is a computer engineer and Ph. D. in Computer Architecture from the Polytechnic University of Catalonia (UPC). Dr. Pastor has been working in the UPC's since 1992 and is currently an associate professor in UPC's School of Telecommunications and Aerospace Engineering in Castelldefels. He is co-founder and head of Intelligent Communications and Avionics for Robust Unmanned Aerial Systems (ICARUS) research group.