Developing a Controller Pilot Data Link Communication Simulator

Rui Ferreira, Manuel Dias, José Vermelho
NAV Portugal, EPE
Firstname.Lastname@nav.pt

António Ferreira
University of Lisbon
asfe@di.fc.ul.pt

Abstract—Radio frequencies for controller pilot communication are becoming a scarce resource due to increasing air traffic worldwide. The controller pilot data link communication (CPDLC) technology, which is already mandatory in new aircrafts, aims at reducing radio interferences and misunderstandings of commands by allowing routine voice conversations to be carried out via text messaging. This paper describes the requirements, architecture, and development of an air traffic management (ATM) system with CPDLC capabilities in the context of a national air navigation service provider. We discuss some challenges and limitations that we encountered, and highlight the importance of building the CPDLC system based upon the existing framework for ATM simulation, which is designed to support on job training and system testing.

Air traffic management, simulator, controller pilot communication, data link, system testing, on job training.

I. MOTIVATION

The volume of air traffic has been increasing worldwide over the past years [1] and this trend is expected to continue in the future at an annual rate of about 3%, meaning that in 2030 there will be almost twice the number of flight movements than today [4]. This situation poses upcoming safety problems to air traffic management (ATM) because radio frequencies used in today’s voice-based controller pilot communication are becoming a scarce resource. One reason for this is that when the number of aircrafts flying in a sector (or geographical area) reaches a limit, the sector is split in two, therefore requiring an extra radio frequency that is assigned to a new controller.

Naturally, a consequence of the scarcity of radio frequencies is that more people have to share the same voice channel, thus increasing the risk of communication problems, which are a significant source of operational errors and pilot deviations, due to noise and interferences, ambiguous wording, message complexity, and other factors [18]. Furthermore, since controller pilot confusions are typically resolved by repeating messages, the tendency is for even worse frequency congestion.

To tackle the scarcity of radio frequencies and increase the operational capacity of the air national service providers, or ANSPs, responsible for the safe, orderly, and rapid flow of air traffic, the International Civil Aviation Organization (ICAO) and the European Commission, through the European Organization for the Safety of Air Navigation (EUROCONTROL), have approved the controller pilot data link communication (CPDLC) technology, which offers text messaging over digital communication channels for connecting aircrafts and control centres via the Aeronautical Telecommunications Network, or ATN [22]. The CPDLC technology should gradually replace some of the routine, non-critical, voice messages exchanged between pilots and controllers during operations on the ground and in the air. The following are some of the main advantages of using CPDLC messages for air traffic management:

- Increased communication channel capacity, because fewer bits are needed to transfer text messages than the equivalent verbal messages [20]. Also, retransmissions requests by pilots or controllers are almost unnecessary, even if the radio channel is noisy, because errors in digital text messages can be automatically detected and corrected [17];
- Increased sector productivity and capacity, due to the possibility of aircrafts periodically transmitting status reports that otherwise would have to be requested and answered via voice conversations [8], and also from the simplification of routine procedures, for instance, by assigning recurrent messages to dedicated buttons [28]. This reduced need for verbal communication has been linked to lower controller workload, meaning that up to about 15% more aircrafts can be safely managed in a single sector [20].

Both of these CPDLC advantages mitigate the scarcity of radio frequencies by better utilizing existing resources, with further optimizations in data link systems indicating that air traffic in Europe should be supported up to the year 2050 [24].

To further emphasize the importance of the CPDLC technology in air traffic management, the European Commission, through the EUROCONTROL, has approved an implementing rule that mandates the installation of digital link systems in all aircrafts built since 2011, and the same applies from 2013 onwards to ANSP organizations that have adhered to the Link-2000+ programme [5].

This paper describes the requirements, architecture, and development of an ATM simulator with CPDLC capability, which is designed for on job training and system testing in the context of a national air navigation service provider. We focus here on a subset of the messages, related to the en route flight segment (high altitude, non-stop, flights), which is less risky compared to landings and take-offs and, thus, more appropriate for the initial operations in the production ATM system.

The paper is organised as follows: in the next section we present the related work, namely other uses of simulators concerning CPDLC, most of which for evaluating network performance; in Section III we describe the framework we use for ATM simulation and in Section IV we explain the integration of CPDLC capabilities in the simulator; in Section V we discuss some challenges and limitations we encountered; and in the last section, we end with the conclusions and future work.
II. RELATED WORK

The CPDLC technology has been maturing over the past years, having captured the interest of researchers who have used computer simulators for a variety of tasks, such as:

- Workload evaluation, based upon the execution of cognitive task models in predefined scenarios of communication, which have predicted imbalances between pilot and co-pilot workload because of the change from the aural to the visual modalities [14] and increased controller efficiency in a mixed visual/aural environment [28];
- Usability evaluation, using high or low-fidelity prototypes that replay air traffic flow scenarios to uncover issues in the human-computer interaction [13], such as the screen getting cluttered when CPDLC capabilities are integrated in a decision support tool [23] or the increased risk of controllers executing redundant actions when text messages replace voice conversations [11];
- Network performance evaluation, in which CPDLC messages are inserted in the communication network to verify that adding a new type of message to the protocol continues to satisfy the overall transfer delay requirements [21], to demonstrate that a protocol is more efficient than another [19], or to show the capacity of a proposed data link configuration can support future traffic demand [24];
- Operational evaluation, accepting inputs from human pilots and controllers during the simulation. We found three such simulators in the literature: Advanced Communications for ATM, AC/ATM [15], User Requested Evaluation Tool with CPDLC, URET/CPDLC [3], and Communication, Navigation, Surveillance for ATM, or CNS/ATM [8]. The three simulators in the previous point are closest to our work as they can potentially be used for both on job training and system testing due to the participation of pilots and controllers. However, to the best of our knowledge, research has focused on network performance and usability evaluations.

As a matter of fact, the stated objective of CNS/ATM is to assess CPDLC in an air traffic network pushed to its extreme conditions [8], and a major goal of AC/ATM is to estimate the number of aircraft that can safely operate on a single frequency, with much emphasis on the simulator supporting up to 160 aircrafts simultaneously [15]. Finally, URET/CPDLC research reports on usability changes due to the integration of CPDLC capabilities in the existing URET decision support tool [3,23].

From this review of the state of the art, we take the opportunity to highlight the importance of researching ATM systems design and development considering the synergies between on job training and system testing, as we describe next.

III. FRAMEWORK FOR ATM SIMULATION

As mentioned earlier, the Link2000+ programme coordinates the application of CPDLC systems in ANSPs from 2013 onwards. But, as new aircrafts now have digital data links, and given the advantages of CPDLC, it is in the best interest of ANSPs to start the development ATM systems with CPDLC capabilities rapidly. In this section we describe work conducted in an ANSP to accommodate the required changes using an in-house framework for ATM simulation, called SIMATM.

A major objective of SIMATM is to leverage the synergy between on job training and system testing [7]:

- On the one hand, trainees have access to a replica of a real ATM system, called LISATM, which is clearly desirable;
- On the other hand, the events generated by the trainees while interacting with LISATM can actually be part of the system testing, complementing static, scripted, scenarios.

In the ATM simulators stated in the related work this synergy is less effective (or non-existent) because human controllers use high/low-fidelity prototypes instead of real systems.

A. Framework Components

SIMATM is a component-based Java framework for ATM simulation and is designed for extensibility by adhering to the model-view-controller design pattern [26]. Fig. 1 shows a data flow diagram based upon the SIMATM architecture specification [16], illustrating its main internal components as well as the interaction with the LISATM external entity.

![SIMATM Components and Interaction with LISATM](image)

Each component in SIMATM is responsible for providing a human-machine interface (HMI) and for assuring the integrity of its own data. Each of the components is described next:

- Scenario Editor: Allows the creation and updating of simulated exercise scenarios. Each scenario is stored in XML format in a database.
- Load Scenario: Converts scenarios into flight plans, which are used by LISATM (and seen by human controllers), and also by the Track Generator and Pseudo-Pilot.
- Track Generator: Calculates the next position of aircrafts considering the information in the scenario and commands issued by the Pseudo-Pilot and Game Supervisor. It also feeds LISATM with radar data.
INTEGRATING CPDLC IN SIMATM

In this section we describe the integration of CPDLC capabilities into SIMATM, focusing on an illustrative subset of the messages, namely the climb/descend clearances that occur during the en route, high altitude, flight segment (see Table I).

The complete set of messages approved for the Maastricht upper area control, which pioneered the use of CPDLC in Europe, additionally includes radio frequency changes, turns and headings changes, microphone checks, and more [25].

<table>
<thead>
<tr>
<th>Direction</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot to controller</td>
<td>REQUEST [level]</td>
<td>(Similar to the messages above, but initiated by the pilot, who must wait for the controller response)</td>
</tr>
<tr>
<td>Pilot to controller</td>
<td>REQUEST CLIMB TO [level]</td>
<td></td>
</tr>
<tr>
<td>Pilot to controller</td>
<td>REQUEST DESCEND TO [level]</td>
<td></td>
</tr>
<tr>
<td>Both directions</td>
<td>UNABLE</td>
<td>Cannot comply</td>
</tr>
<tr>
<td>Both directions</td>
<td>STANDBY</td>
<td>Further message will follow</td>
</tr>
<tr>
<td>Controller to pilot</td>
<td>MAINTAIN [level]</td>
<td>Maintain [level] altitude</td>
</tr>
<tr>
<td>Controller to pilot</td>
<td>CLIMB TO [level]</td>
<td>Climb to [level] altitude</td>
</tr>
<tr>
<td>Controller to pilot</td>
<td>DESCEND TO [level]</td>
<td>Descend to [level] altitude</td>
</tr>
</tbody>
</table>

From the subset of CPDLC messages in Table I we enumerate functional and non-functional requirements that lead to modifications in some of the SIMATM components. However, before that we introduce the software development process that we follow in-house.

A. Software Development Process

The project to integrate CPDLC capabilities in SIMATM, much like our other assignments to build complex and critical systems, follows a documented software development process, which is based upon the V-model [2] complemented with good practices from agile methodologies [9,10].

In a nutshell, besides the expected division of the process into phases (namely the requirements gathering, architecture, detailed design, coding, unit testing, integration testing, and validation), in which the outputs of each phase feed the inputs of the next ones, we also identify the functionalities that are more likely to suffer modifications.

Then, for each of these individual functionalities we apply a complete development cycle to ensure that what is built is what the client specified and so that the client can periodically validate our work. This means the system is developed incrementally to guarantee that the adaptation to changes has a minimal impact on the project planning and satisfies the client.

Next, we show outcomes from the requirements gathering phase, specific to the messages in Table I, and considering the context provided by the SIMATM components in Fig. 1.

B. Functional Requirements

The integration of CPDLC capabilities in SIMATM determines three functional requirements that should be satisfied during both the setting up and execution of an ATM exercise:

- The Scenario Editor should be able to create an exercise scenario containing aircrafts with CPDLC capabilities;
- The Pseudo-Pilot should be able to see if an aircraft under his/her own control has CPDLC capabilities;
- The Game Supervisor should be able to see, at any instant of the exercise, if a specific aircraft has CPDLC capabilities, and check if they are currently active.

Next, we present the behaviour that should be satisfied only while an exercise is running (not paused nor being setup):
• When SIMDLS receives a climb/descend clearance from LISATM, it should forward the information contained in the message to the Pseudo-Pilot, which presents it in the HMI. SIMDLS should allow the Pseudo-Pilot to respond using WILCO, STANDBY, and UNABLE messages;

• SIMDLS should allow Pseudo-Pilot to send REQUEST type messages to LISATM and forward the controller’s reply, UNABLE or STANDBY. It should also allow the Pseudo-Pilot to send a final reply, WILCO or UNABLE.

Before the messages in Table I can be processed by SIMATM, and especially by LISATM, in which controllers use real equipment, there are session initiation and termination messages (similar to logons and logoffs) that must be processed to determine if the CPDLC capabilities of an aircraft are currently active. The details of these were omitted for simplicity sake.

C. Non-Functional Requirements

Besides the behaviour allowed by SIMATM it is important to understand the qualities inherent to the operation of the system. To this end, we first introduce two types of timeouts that can occur in a CPDLC context:

• Operational Timeout: When the Pseudo-Pilot does not respond with a WILCO or UNABLE message within 120 seconds after the controller in LISATM has sent a message to the aircraft, or 120 seconds after the controller has received a STANDBY message from the Pseudo-Pilot;

• Controller Timeout: When the controller in LISATM does not respond with at least an UNABLE message within 120 seconds after a REQUEST message initiated by the Pseudo-Pilot, or after 120 seconds upon sending a STANDBY message to the Pseudo-Pilot.

The clocks watching both types of timeouts are reset when a STANDBY message is received. From the two definitions of timeout, it is now possible to fully describe the non-functional requirements of a CPDLC-enabled SIMATM:

• Performance: Messages should take less than one quarter of a timeout duration (operational or controller) to be delivered from LISATM to the Pseudo-Pilot and vice-versa;

• Reliability: When a message is incorrectly received or a timeout happens, SIMATM must consider the data communication link is corrupt and should shut it down;

• No Duplication: Messages sent by LISATM should be received only once by SIMATM, irrespective of the noise in the radio communication channel;

• No Creation: Messages should be uniquely identified, and also contain the identifier of either the aircraft or the controller that originally created the message; in other words, extra messages should not be created;

• Integrity: Messages received from or sent to LISATM should not be modified by any SIMATM component;

• Usability: The interaction with the HMI of the Pseudo-Pilot must be accomplished in at most four steps, and the previous steps should always be indicated so that the human pilot can rapidly restore the context; in addition, after the insertion of a command by a pilot, SIMATM must ensure the performance requirement is accomplished;

• Capacity: SIMATM should provide all requested CPDLC capabilities to aircrafts participating in an exercise.

All requirements for CPDLC were defined in terms of the SIMATM framework and its interaction with LISATM, which naturally lead to changes in the existing components.

D. Changes to the SIMATM Components

In order to satisfy the identified requirements it was necessary to add CPDLC functionality to several of the SIMATM components (see Fig. 1 for reference).

Starting with the Scenario Editor, it has to support the creation of scenarios with aircrafts having CPDLC capabilities. This involved modifying the representation of the details of a scenario in the database and the corresponding changes to the HMI, such as the creation of new forms to be filled out.

Regarding SIMDLS, it handles all data link transfers between LISATM and SIMATM, and so an upgrade to CPDLC messaging was needed, namely to convert the new messages in ASN.1 format to xmlRac and vice-versa. In addition, new types of log entries were defined so that all message exchanges could continue to be recorded. This log is used to facilitate the detection of software errors and also to make it possible for the Game Supervisor to gather statistics about the types of messages most frequently used during an exercise.

Another responsibility of the Game Supervisor is to monitor the exercise, and in this matter s/he can inspect the CPDLC capabilities available in each aircraft using the HMI.

Finally, the Pseudo-Pilot component needed several new functionalities, especially in the HMI, which features additional menus and forms for creating and responding to CPDLC messages, and also to initiate and terminate data link sessions (briefly mentioned in Section IV.B).

V. DISCUSSION AND LIMITATIONS

We now discuss some challenges and limitations encountered during the integration of CPDLC capabilities in SIMATM. We begin with an explanation about the advantages and disadvantages to system testing of having the SIMDLS component assume LISATM is an external entity, when, in fact, it is the same ANSP that develops and maintains both systems. Then we describe the implications of the ICAO 24-bit aircraft identifier [6] having so far being overlooked, and how that affects on job training.

A. Communication between SIMATM and LISATM

As explained earlier, the SIMDLS component is responsible for forwarding messages between the SIMATM infrastructure and the LISATM external entity. In doing so it needs to convert messages from ASN.1 to xmlRac formats, and vice-versa (see Fig. 1). Alternatively, and since we have access to the LISATM internals, which actually uses the same xmlRac protocol for its inter-component communication, we could have avoided the data conversion altogether by directly linking SIMDLS to a so-called data link processor (or DLP) in LISATM, which is itself very close to the HMI operated by the human controller. This approach would simplify the design of SIMDLS and reduce the complexity of implementing CPDLC messages to only one format.
However, the main disadvantage of directly linking SIM- DLS to an internal component of LISATM is that integration testing and validation would not stress the full data link connection that exists between real aircrafts and LISATM, which relies on messages in ASN.1 format. In addition, the SIMDLS data link simulator represents all-purpose communication channel defined in the ATN [22], so it goes beyond CPDLC messaging. Again, it is advantageous to test the entire chain of components for the other types of messages and services proposed by the air traffic management industry.

B. Lack of Unique Aircraft Identifier

An important element of ATM simulation scenarios is the flight plan, which, among other blocks of data, describes the time of departure and arrival from/to airports, and, particularly, the aircraft identification, usually in the form of a callsign. The traditional format of a callsign is two or three letters followed by the flight number (which controllers always use to refer to an aircraft in voice-based communication), but recently ICAO has recommended the adoption of a 24-bit identifier to guarantee that only one aircraft receives its designated CPDLC messages, even if it happens that the same callsign is wrongly used by two airborne aircrafts [6].

However, as it currently stands, SIMATM does not feature the ICAO 24-bit aircraft identifier, even more so because the related LISATM flight data processing system (FDPS) is still being upgraded, a process that started after the 24-bit requirement became stable.

VI. CONCLUSIONS AND FUTURE WORK

The integration of CPDLC capabilities in an ATM simulator facilitates on job training for novice and expert controllers in the upcoming transition from voice to text-based communication with aircraft pilots, essential to ease the scarcity of radio frequencies. To this end, we described the modifications made to the components of an in-house simulation framework, SIM- ATM, and highlighted the importance of running and testing it together with a real air traffic management system.

Our current priority is to incorporate the ICAO 24-bit aircraft identifier in the flight plans and in the HMIs of the relevant components, as this improves the realism of the exercise scenarios; in fact this will be a feature of the next version of SIMATM. We also plan to go beyond the en route flight segment and into the more challenging airport approach and ground operations, but that will happen further down the road.

REFERENCES