



Development of an ATC Tower Simulator to Simulate Ground Operations

Zarrin K Chua, Fabien André, Mathieu Cousy

► To cite this version:

Zarrin K Chua, Fabien André, Mathieu Cousy. Development of an ATC Tower Simulator to Simulate Ground Operations. AIAA Aviation 2015, AIAA Conference on Modeling and Simulation Technologies, AIAA, Jun 2015, Dallas, United States. 10.2514/6.2015-2494 . hal-01174666

HAL Id: hal-01174666

<https://enac.hal.science/hal-01174666>

Submitted on 9 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Development of an ATC Tower Simulator to Simulate Ground Operations

Zarrin K. Chua*

Institut Supérieur de l'Aéronautique et de l'Espace - Supaero, Toulouse, France

Fabien André[†] and Mathieu Cousy[†]

Université de Toulouse, ÉNAC, Toulouse, France

The expected growth in air traffic has resulted in many projects attempting to mitigate the problem of air traffic control, particularly at airports. These projects have focused on characterizing the workload of the air traffic control officer and/or introducing automated technology to assist in his or her responsibilities. Many of these projects have used existing simulation facilities and technology or developed analogous environments to simulate the air traffic control tower. However, such high-fidelity simulation facilities are in limited locations or not easily accessible. Results gathered in simplified environments, while accurate in a controlled setting, may be limited in their extrapolation to real-world applications. Furthermore, little guidance is provided in the literature for developing an ATC tower micro-world, particularly in terms of knowing which elements are most necessary to simulate and the most effective way to actualize such functions. In this paper, we present the development of an air traffic control tower simulator, specifically for the ground controller, intended for the design and validation of potential automated concepts. We discuss our reasoning for the simulation of specific elements, the hardware and the software used, and our methodology for simulating ATC operations within the tower. This paper concludes with lessons learned and results from the simulation validation, a novel 10-question 5-point Likert scale questionnaire.

I. Introduction

Improving air traffic control (ATC) operations around the airport is a method for reducing delays and maintaining safety, especially in light of the predicted increase in air traffic. However, it is difficult to test new concepts with real-time operations, thus prompting the need for micro-worlds or simulated environments. Such simulated environments may have various levels of fidelity, ranging from evaluating a singular tower position with a limited number of traffic or including every detailed aspect of airport operations from gate to runway and beyond. However, many of these simulators require the use of multiple experts (pilots or other air traffic control officers, ATCOs) and proprietary software. Additionally, these simulators are not easily accessible due to their unique use for training and advanced concepts testing. Therefore, most tower research activities are limited to select research institutions or are performed with a low fidelity simulator that may bias measured performance. Greater diversity in the field of tower operations research could be achieved with a simulator that is capable of achieving short-term aims while providing the flexibility for continued long-term validation improvement. Planning for both short- and long-term and compensating for a lack of expert pilots or air traffic controllers does require an understanding of what can and should be simulated, with an idea of how such a simulator could be quantitatively validated.

In this paper, we discuss the development of a medium-high fidelity tower simulator developed from open-source software that uses non-pilot personnel to respond to air traffic controller commands. We begin by

*Post-Doctoral Researcher, Département Conception & Conduite des Véhicules Aéronautiques & Spatiaux, Édouard Belin, 31055 CEDEX 4, Toulouse, France

[†]Research Engineer, Aeronautical Computer Human Interaction Lab, 7 avenue Édouard Belin, 31055 CEDEX 4 Toulouse, France

highlighting several well-known existing tower simulators and examples of tower simulators in the research. Section III discusses the different simulation elements that could be simulated, including a breakdown of which elements are of particular importance to specific tower controller roles. Next, we present the hardware and the software used in the development of our simulator. Section V consists of a thorough discussion on the use of pseudopilots, simulation alternatives, and training of non-pilots. Lastly, this paper concludes with feedback, informal and formal through a novel questionnaire for simulation validation and the results specific to our simulator and two others in France.

II. Literature Review

There are several existing facilities around the world that simulate the ATC tower. Since most aviation administrations in each country offers at least one ATC tower simulator, it is impossible to describe each and every one in this paper. Notably, the simulators most commonly used in the literature are from the FAA, NASA Ames, and the MITRE Corporation. All of these simulators employ pseudopilots, or researchers/experts that react in real-time to the participant's commands, directing aircraft and responding using the correct lexicon. This paper will refer to this type of experiment role as *pseudo*, as to avoid confusion when discussing actual pilots.

The FAA employs the Tower Simulation System,^{1,2} which is present at most major airports in the United States and the FAA Training Academy. This system is capable of handling three tower positions and supporting several pseudos. The NASA Ames' Future Flight Central (FFC)^{3,4} is a tower simulator that is capable of simulating a 360° window view of an airport. This simulator can simulate up to 12 controller positions, 8 ramp controllers, and 13 pseudos. Verma et al.⁵ used the FFC simulator in their project for testing their proposed taxiing operation tool. They generated traffic using the Airspace Traffic Generator and a simulator of the Airport Surface Detection Equipment to monitor aircraft positions on the surface. The MITRE Aviation Integration Demonstration and Experimentation for Aeronautics (IDEA) laboratory is also equipped with a tower simulator,⁶ providing a 200° view from the tower and supporting various controller roles and pseudos. Stelzer, Stanley, and Stanley⁷ simulated Dallas-Fort Worth (DFW) using the MITRE IDEA laboratory in their study on surface trajectory-based operations concepts. Their simulation featured electronic flight strips and two pseudos, to handle both the local and ground positions simultaneously.

Other research in tower operations research has used simulators and simulation software from around the world. The École Nationale de l'Aviation Civile (ENAC) in Toulouse, France is equipped with Scan-Sim,⁸ which offers up to a 360° view from the tower and is capable of simulating several airports around France. Eurocontrol's ESCAPE (Eurocontrol Simulation Capability and Platform for Experimentation) simulator can handle as many as 70 pseudos to manage aircraft for a variety of different air traffic controller positions.⁹ Martin et al.¹⁰ used the Multi-Aircraft Control System (MACS)¹¹ simulation to simulate Los Angeles International Airport (LAX) for the final, feeder, and en-route controller positions. The Standard Terminal Area Replacement System and the Display System Replacement were emulated and incorporated in the Terminal Area Precision Scheduling and Spacing system that they were testing. Jakobi, Teotino, and Montebello¹² performed their experiments in the Apron and Tower Simulator in the generic experimental cockpit simulator at DLR-Braunschweig. This simulator provides a 360° window view and incorporates the hardware and software used at Prague Ruzyně (PRG), including a Traffic Situation Display and electronic flight strips. The ATCOs were also equipped with a higher level Advanced Surface Movement Guidance and Control System (A-SMGCS) that was under consideration for future tower operations. Straussberger and Schaefer¹³ developed a simulator on the premises of a "European Air Traffic Control Center" that allowed for differing traffic variations of arrivals and departures. No other information was provided regarding the simulator itself. Small, Hammer, and Rouse,¹⁴ in their preliminary study on ATC symbology, developed a simplified tower simulator that contained a single screen with radar information. They noted that continuous monitoring of a display is what occurs during low visibility operations. Additionally, the participants did not control the aircraft. Small, Hammer, and Rouse noted that while such control would increase the simulation fidelity, it would also result in differences between participants, thus losing comparability between subjects. Additionally, the scaling of the radar screen was adjusted to account for eye tracker use.

Despite the impressive simulation capability offered by each of these simulators, they are not commonly available for general usage.¹⁵ Many of them are in active use for training ATCOs and cannot be easily spared for external research. Additionally, as the manpower required to simulate a single scenario is unfavorable (at least 1:1 in terms of pseudo to participant, the pseudos being trained aircraft operations personnel),

it severely limits the potential for tower-based research projects at other institutions such as those at the university level. However, as we outline in this paper, it is possible to develop a reasonably medium-high fidelity scenario using non-pilots and open-source software for specific project needs. Such a simulator can be adapted and worked upon for long-term improvement while reaching short-term project goals.

III. Methodology

The number of variables that can be simulated within ATC tower operations is large, interrelated, and primarily based on the intended usage of a simulator and the degree of fidelity required. In general, there are five categories of different elements that can be simulated: external infrastructure, operations, aircraft, work environment, and environment. Collectively, there are 51 different elements, although this number may vary more or less depending on breakdown and definition. The list in Table 1 is meant to serve as a initial starting point in considering which elements could and should be modeled.

Table 1. ATC Tower simulation elements

<u>External Infrastructure</u>	<u>Operations</u>	<u>Aircraft</u>
Airport	Number of aircraft	Arrival/Departure
Airport size	Tower call frequency	Aircraft type ●★
Sector within airport	Aircraft size diversity	Aircraft size ●★
Surveillance zone	Airline diversity	Airline
Buildings ◇★	Ground dynamics ★	Pilot Communication profile
Service roads ◇★	Taxiway restrictions ★	SID routes ●★
Service vehicles	Noise restrictions	Airport origin/destination ●
Technology accuracy	Technology usage diversity	Transponder code
Runway entry/exit ramp ●★	Parking area-based responsibilities ★◇	Technology usage
Parking area entry/exit ramp ★◇	Alarms (i.e. runway incursion) ●★	CTOT ★
Sector frequency number	Configuration change ●★	EOBT ◇
	Arrival/Departure diversity ●	TOBT ◇
	Parking stand availability ★◇	TSAT ◇
<u>Work environment</u>	<u>Physical Environment</u>	<u>Legend</u>
Number of other controllers	Precipitation ●★◇	● Local controller
Types of other controllers	Visibility ●★◇	★ Ground controller
Tower supervisor	Wind ●★◇	◇ Apron controller
Ground radar ★ ^a	Time of day ★◇	
Radar ●		
Window view ●★◇		
Future technologies		
Accuracy of equipment		
Arrival/Departure managers		
Radiotelephony ●★◇		

III.A. General simulation considerations

The external infrastructure consists of the physical aspects of the airport, such as the airport, the sector, the buildings, and the technology present at the airport itself. Depending on the simulator, one may choose to model a specific existing airport, or model a variety of different airports or airport sizes (i.e., number of runways). Choosing to model a specific airport will dictate the relevancy of the other elements on the list. The studied user also determines which airport sector is of interest, and what aspects of this sector (and other bordering sectors) are within the user's surveillance zone. Not all elements within the studied sector need to be simulated. For example, large airports may feature several runway and parking entry/exit ramps that can be simplified to one per runway or parking area. This simplification may also accommodate for participants unfamiliar with the particular airport layout.

Operations consists of movements around the airport, including the aircraft fleet (number, tower call frequency, diversity in size, airline, dynamics, arrival/departures). This category also covers airport-specific rules, such as taxiways restricted to heavy or forbidden to aircraft of a certain wingspan, or parking areas where pushback is under ground controller responsibility. Technology usage diversity includes the different types of technologies used by the individual roles, whether automation has been added specifically for one controller but not the other. Aircraft ground dynamics may include a variety of assumptions, ranging from different velocity profiles based on aircraft size, maximum turning velocity, and non-constant acceleration. Modeling operational events such as traffic load or configuration changes can modulate task difficulty for ATCO.¹⁶

^afor large airports

The category of aircraft elements spans all of the characteristics with respect to a single aircraft, effectively the information on a paper flight strip. Additionally, we include aspects such as pilot communication profile (accents, cadence, language) and pilot competency, which affects pilot error (runway incursion, deviation from assigned commands, etc). Flight plans such as airport origin/destination and Standard Instrument Departure (SID) routes also add to the realism. Lastly, technology usage is a new aspect, to reflect for instance the introduction of Datalink, thus changing the manner of communication with the tower. Diversifying the aircraft not only adds to the realism, but also can introduce sub-problems to the ATCO role and or lend focus to research.

The physical environment describes the precipitation, visibility, wind, and time of day of the simulation scenario, all of which may play an impact on the ATC task. For example, snow can lead to deicing procedures, heavy fog requires low visibility procedures (LVP), the airport configuration may change due to strong winds, and airport operation procedures are different at night versus the peak point of the day.

The work environment is the working space of the ATCO. The simulation may opt to model other ATCOs (or have several participants in several roles simultaneously) or the Tower supervisor. Unlike the other categories, these elements are primarily hardware rather than software, which may contribute to development time or appropriate workarounds. For example, a flight strip printer is not a commonplace object. As such, personal printers may need to be adapted to print the same physical dimensions, or the physical strips printed prior to the simulation run.

III.B. Role-specific considerations

In this next section, we discuss each of the three roles individually, with the acknowledgment that at certain airports, these roles are often conducted by the same person. While all of the elements in Section III.A are important to modeling any of these roles, there are certain pieces of information that are more critical to those than others (Table 1). However, the importance of these elements to the specific roles are forever debatable and their inclusion in the simulation is ultimately driven by the research needs. Nevertheless, this section acts as a starting point for this critical discussion.

The local controller (indicated by a bullet • in Table 1) is primarily focused on landing and take off, with a goal of optimizing runway usage. As such, he primarily relies on details such as aircraft size, aircraft type, SID routes, and airport origin/destination (which dictate the SID route). He uses the radar to monitor final leg and build the runway sequence. This controller must also be aware of runway incursion by pilots or service vehicles, particularly during moments of high congestion. Similarly, low visibility reduces his ability to see aircraft through the window view and limits the runway usage.

The ground controller (noted as a star ★) is primarily focused on the sector between the runway and parking. The physical environment has a significant impact on his performance, with low visibility procedures limiting the maximum taxiing speeds and deicing stations causing blockages near the runway. Large parking areas may also include multiple entry and exits, with specific gates or areas requiring ground controller clearance for pushback. Pilot errors around the airport may require tractors to disengage from a taxiway erroneously taken. The ground controller may also have different responsibilities depending on the time of the day (i.e. responsible for the apron sector during early morning, but not during the day¹⁷).

The apron controller (marked with a diamond ◇) is primarily focused on the parking areas, with the assignment of parking gates, pushback, and managing service vehicles. His sector has him in closer proximity to the airport buildings and the service roads. This controller is also the first step of the departure scheduling process and thus must ensure that aircraft pushback at the appropriate time to meet their calculated Take Off Time (CTOT).

We do not address in this paper Clearance Delivery controller and possible flow position. We consider this as input of our system (with direct constraint and consequence, such as the time an aircraft will contact apron, or a target take off time to honor). All of these roles might be each shared by different people or performed by one person, adapting to traffic.

IV. Hardware & Software

This section describes the rationale of choices and tracks explored while developing this specific simulator. If a generic and high-end simulator appears to be an investment for a national research center,⁴ this paper serves to be a blueprint and a guide to the design and assembly of a budget simulator to meet those needs.

Most importantly, the system described in this section is highly configurable and modular, thus capable of being easily adapted to different research needs. The process described does not introduce new simulation techniques or methodologies, but rather is a composition of existing and general public components. The simulator featured in this paper is based on the airport operations, specifically for the ground controller, at the south end of Roissy Charles-de-Gaulle (CDG) in Paris, France. It was built to test new technologies and automated concepts for improved taxiing operations. Figure 1 presents the simulation schematic of all the major elements. Briefly, CDG has two parallel runways (27/09 L/R) on north and two on south (26/08 L/R); has a north and a south tower; and separates the local, ground, and apron controllers for each end.

With respect to the previous section, 43 of 51 variables were included, with eight excluded: arrival/departure managers, noise restrictions, precipitation, time of the day, pilot accents and cadences, transponder codes, CTOT, and service vehicles. While tractors are simulated, we do not include other vehicles such as fuel trucks, baggage handling, and passenger buses. Transponder codes were not included in our simulator as no light aircraft fly from CDG. While our simulator was capable of simulating any time of day, our traffic patterns did not rigidly respect those of early morning at CDG. Our pseudos spoke in both French and English, but global accents were not an emphasis in this simulation. The exclusion of these elements were carefully selected based on the priorities and research aims.

Prior to simulator development, several commercial tower simulators were considered before choosing to develop one from scratch. As noted in Section II, several simulators were unavailable due to physical distance and limited access to non-training applications. Three commercial simulators were of particular interest: ATTower,¹⁸ Scansim,⁸ and the French Air Navigation Service Provider (ANSP) research center. ATTower (used by the CDG training center and Eurocontrol Roissy simulator) was competitive, Scansim, at the time of this investigation, did not have a model of CDG, and the French ANSP research center CDG simulator was aging and unmaintained. We finally rejected these commercial simulators because of the price and the delays required for significant modifications and additional development. The 30-month project required rapid support of autonomous taxibots^b, a new type of service vehicle that has been proposed for future operations. Taxibot dynamics and 3-D models were necessary for this work, as well as map modifications, detailed simulator logs, and service roads for the taxibots. We also aimed to facilitate the dispersion of the simulator across project partners, and eventually beyond, through an open-source project. This choice of an in-house simulator was further assured with the acknowledgement of the team's previous experience in simulating en-route operations.

IV.A. Operational data

In all cases, the first natural choice is to determine which airport(s) will be simulated. The tower position on the airport will have consequences on external view requirement: a central tower might require 360°, whereas for one on the side of the field, only 180° would suffice. The height and desk setup of the actual tower could also add constraints on the vertical field of view. Depending the specific simulation needs, one could choose to simulate only a subset of the controller positions described in Section III.B.

The availability of airport technical data such as traffic, cartographic, and operational data must be considered during the selection process. There are several publicly available data sources, under different licenses: X-plane data (community feed, GPL,¹⁹ also used by Flightgear) that describe the taxiway network of most large airports, or OpenStreetMap (community feed, ODbL²⁰). The volunteered geographic information description quality may vary but can reach good quality,²¹ albeit depending on community good will^c. More traditional data could be available from ANSP or airport operator which might provide (to anybody or on demand) Airport Mapping DataBases (AMDB). There are also several commercial aeronautical data providers. One can reasonably hope that with the spread of A-SMGCS and ground moving maps, taxiway networks will become more common and easier to obtain in future years.

This simulator used SIA^d AMDB topographic survey of CDG. However, an unexpected issue related to the process of production of this data arose. AMDB Taxiway centerline elements that this simulator uses to extract the taxiway network were traced from orthophotography and thus reproduces the painting flaws: non-joining lines, holes in the lines crossing other paintings (such as runway numbers). It became necessary

^btugs that tow aircraft to the runway threshold, thus reducing reliance on aircraft main engine taxiing

^cfor instance, Vienna International Airport (LOWW) in Vienna, Austria was very accurately modeled with taxi lanes, holding points, disused taxiways, etc. thanks to the initiative of a small group, provided by the airport operator's map.²²

^dFrench Aeronautical Information Service

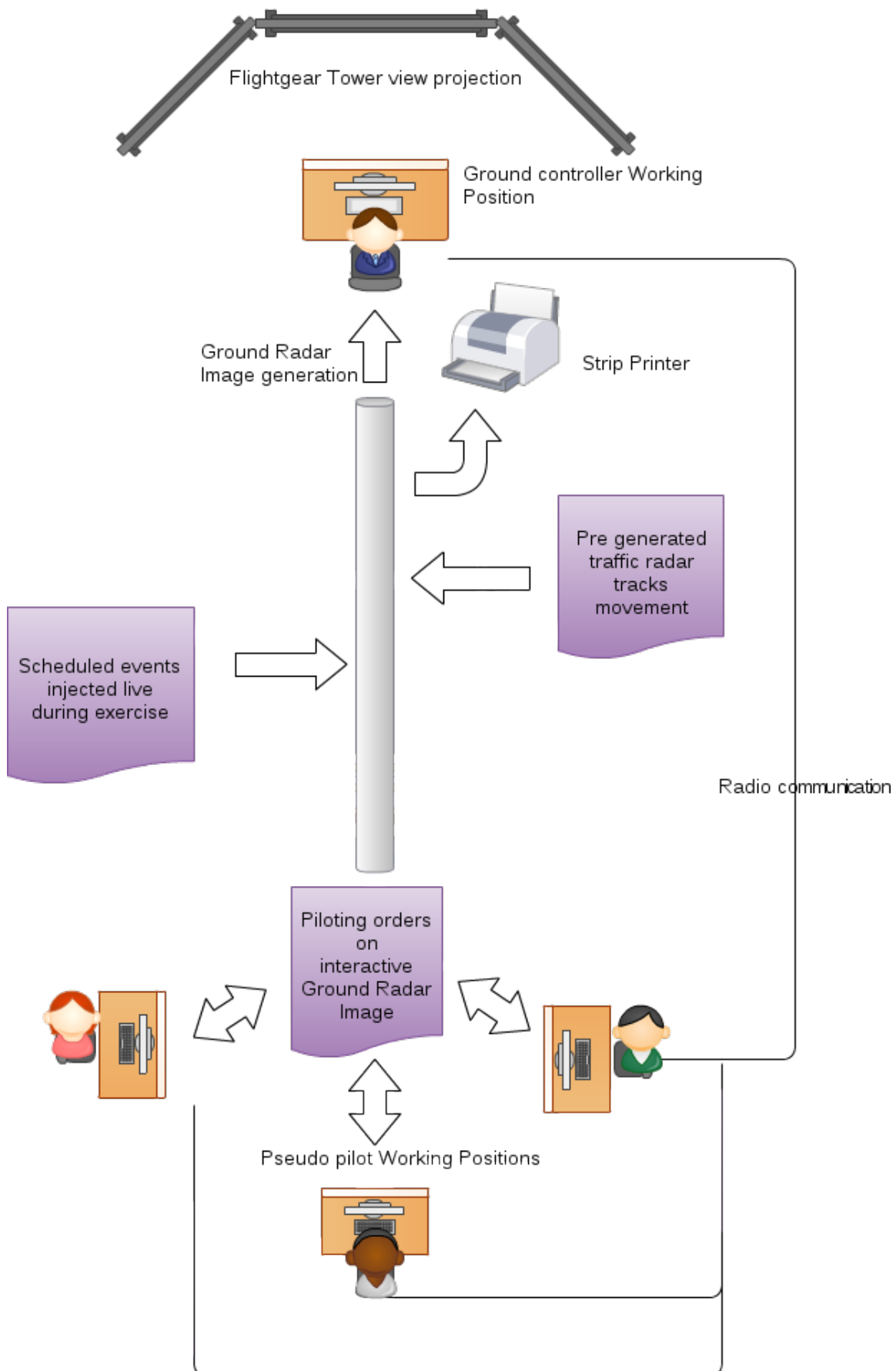


Figure 1. Overview of the simulator

to another tool to merge adjacent extremities (automatic merge based on fix threshold was not possible since the errors exceeded the aspired precision grain).

One of the outstanding parts of a tower simulator is the out-the-window view of the airport. It is paradoxically not so helpful for experimental simulations, since ATCOs spend most of their time head-down staring at a strip board or ground radar images if provided with one (20% head-up at Roissy²³) and even moreso in heavy traffic situations, or in LVP. It is often these situations that are the most studied. The open source flight simulator Flightgear²⁴ was chosen in the end. It provides a sufficient network of most large airport taxiway networks, particularly for CDG^e, and the flexibility of the software permits customization. The multiplayer protocol was used to insert the simulated traffic into the Flightgear framework. The well-detailed documentation provided sufficient guidance to allow for the development of an agent that emulates as many multiplayer instances as the number of aircraft in the simulation.

However, the counterpart of using a flight simulator as an out-the-window viewer was that several unanticipated features required further development. For example, it was necessary to interconnect with the Flightgear Digital Elevation Model (DEM) to seat the aircraft on the ground^f. In addition, there is no assurance that the AMDB routing data and the Flightgear taxiway descriptions match each other. In this particular case, there is not a perfect alignment due to the recent re-work on the CDG taxiway routes, but this discrepancy is hardly noticeable from the tower point of view.

Throughout this process, it is important to note the number of functions and tasks that can be automated, thus allowing for the relatively easy inclusion of other airports. As with all programming projects, developing additional tools robust enough to handle a wide spectrum of cases does add to development time. These concerns should be carefully considered with respect to the project goals, manpower, and available resources.

IV.B. Architecture

The simulation software is anchored around Ivy,^{25,26} a text-based communication bus. It is a central hub to which all individual aspects of the simulation send and receive data, including event- or time-based actions. A distributed architecture of several independent and reusable agents was used, as represented in Figure 2. This simulation is composed of three main components:

- **Ready to Taxi (RTT)**, the simulation engine, starts with a file of pre-recorded traffic. This block manages the simulation clock, displaces aircraft (initial movements recorded at the start of the scenario, then handled via pseudo orders), and delivers other data as flight plans.
- **Airport Human-Machine Interface (HMI)**, a ground radar view of all aircraft plus a flight list of departing flights (*Départ d'Information de Supervision et de Clairance pour les Utilisateurs dans les approches*, DISCUS). A perfect correlation was assumed, providing updates of all vehicle positions at 1 Hz.
- **Scenario Scheduler**, the events coordinator. It can virtually send any message on the bus at a given time.

Collectively, these components manage all of the other simulation aspects, such as the individual pseudo HMI that direct the aircraft around the airport or the flight strip printer. A monitoring agent also analyzes live traffic and logs aircraft start/stop and enter/leave actions within predefined areas. All messages exchanged on the bus are recorded, which allows for replay of the specific exercise *post-hoc*, and a subset of these messages are piped to a spreadsheet for further analysis.

There are two others communication streams involved in the simulation: the traffic injection to Flightgear through the multiplayer protocol and the radio communication system between actors of the simulation (Figure 1). We emulate radio communication with general public off-the-shelf VoIP software^g. These software offer, with minor configuration, several chat channels for radio frequencies, as well as whisper-mode for telephony / direct-communication (with adjacent positions). We connected ATC Push-To-Talk hardware (button and pedal) to this whisper-mode through key-emulation (push-to-talk actions for instances are also logged on Ivy bus).

^eThe simulation scenery for CDG also includes detailed 3-D models of the buildings, which is not the case for every airport.

^fPrior to the resolution of this issue, all of the aircraft rested at the average altitude of the terrain, with a few floating above this threshold and others with gears embedded in the terrain.

^gWe used TeamSpeak.²⁷ Mumble²⁸ could be a free software alternative

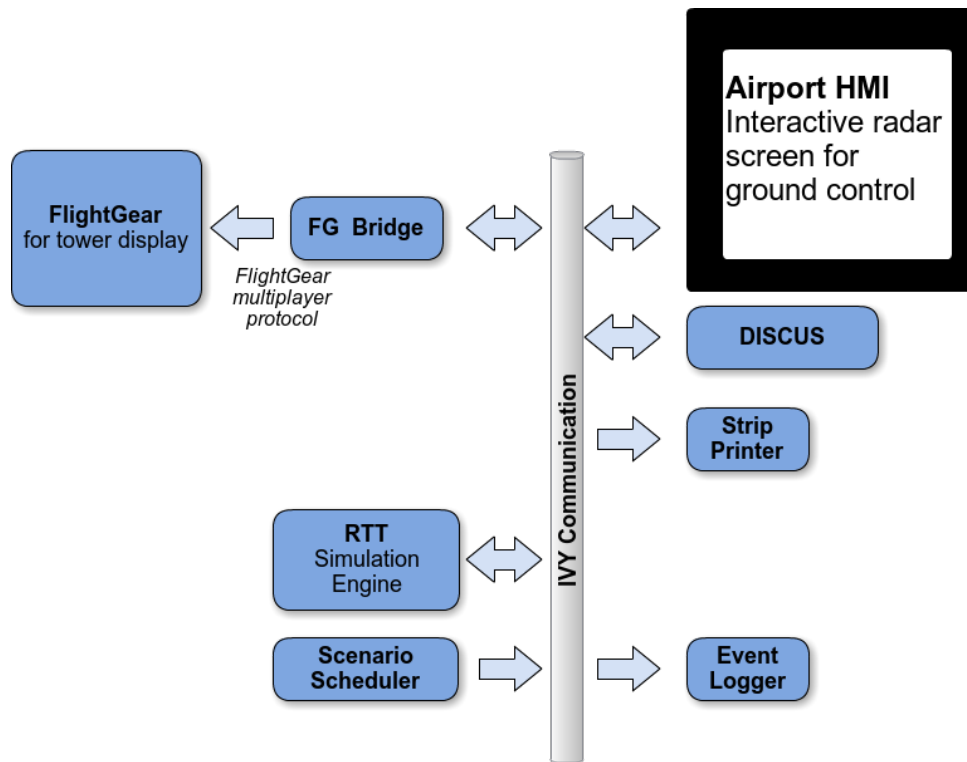


Figure 2. Software Architecture of the simulator

IV.B.1. Ready to Taxi

The ground traffic simulator was developed based on previous work for en-route simulations. Several main additions was to introduce altitude (instead of and with higher precision than flight-levels), to add the attitude (yaw, pitch, roll) of the aircraft (required for the out-the-window view), and to handle new commands (pushback, taxiing, stop/resume) with specific ground physical dynamics. The addition of new data not present in the radar track prevents us from using actual traffic records as an input in the simulation. For this same reason, a trajectory matching strategy (search for a similar, i.e. same path and same aircraft, trajectory in a database) could not be applied to respond to taxiing orders. Furthermore, taxiing is subject to greater speed variation (pilot wanting to gain or lose time) and more conflicts with other aircraft than cruise flights.

The traffic simulator converts pilots orders to 4-D trajectories and the corresponding positions are emitted on the bus every second. Speed according to aircraft and taxiway types are taken into account in this computation.

IV.B.2. Human-Machine Interface

The project is focused on the Ground controller position, hence, a greater emphasis was placed on the development of the ATCO HMI and the interactions needed to input a taxiing route for each aircraft. This interface consists of a radar image showing aircraft track positions on a map of the airport infrastructure. In parallel, the platform also requires a HMI for the pseudo to move the aircraft according to the ATCO clearances. As the same information (e.g., aircraft position, surrounding traffic) and actions (e.g. route inputs, start/stop) are needed on both interfaces, the possibility of using the same tool for both positions seemed obvious. Nevertheless, the constraints on both tools are quite different, for two reasons:

- The route input interaction needs to be different: the ATCO will have an assisted input system that allows a quick definition of standard routes, whereas the pseudo will have a free draw input system that allows simulation of any events such as a wrong turn or an engine failure.
- The messages sent by the HMI must be tagged as ATCO or pilots. ATCO orders may be stored for

other tools such as monitoring clearances but they have no impact on the simulation engine trajectory generation, whereas the pilot's order will be directly interpreted by the RTT component.

IV.B.3. Scenario Scheduler

The Scenario Scheduler component plays the role of the conductor of an orchestra. The operation is straightforward - it reads a scenario file that contains timestamped Ivy messages and send them to the bus when the simulation timer reaches the timestamp. If the message definition is correctly structured and finely detailed, this system permits simulation of a wide range of situations. For instance, this component is used to simulate the departure manager by sending messages to regularly re-order the departure sequence on the ATCO display. Every automatic action can thus be easily implemented. In the same manner, the strips can be printed in the correct timing without simulating a complete ATC system. Basically, during a simulation the Scenario Scheduler sends the following orders:

- Print strip
- Change frequency status of an aircraft (it simulates adjacent sectors radio transfers)
- Update departure sequence
- Update aircraft flight plans (parking information, SIDs, etc)

The Scenario Scheduler component is used both online during the simulations and offline during the preparation of the scenarios. Since the simulation engine is able to generate departure and arrival aircraft upon reception of an Ivy message, the Scenario Scheduler is used to produce the desired traffic load for the different scenarios via two orders: generate an arrival aircraft on a given runway and exit taxiway; and generate a departure aircraft at a given gate. The generated radar tracks are then dumped into a traffic file that the simulation engine is able to replay.

This component allowed for the simulation of a change in configuration during a scenario. The Scenario Scheduler reacts on a specific Ivy message that is manually sent by the simulation coordinator to notify the actual configuration change time. On this notification, the Scenario Scheduler loads a new scenario file which describes arrival flights in the new configuration and updates departure flight plans information with the new departure runway.

IV.B.4. Physical setup

The simulator is deployed on general public hardware. The computer managing out-the-window view is an high-end game-oriented computer plugged to three projectors (Figure 3). Projections screens consist of three wood frames covered with PVC fabric built in-house. The ground radar image and pseudo positions run on average desktop computers. Pseudos are in the same room as the tower simulator, but are far enough not to be heard without radio (Figure 1).

Actual ATC hardware such as stripboards, a strip printer, and radio equipment may be replaced by simplified versions. For example, the strip printer may be replicated by pre-printing a stack of strips and playing a printing sound at the intended time. Even a small cardboard box would suffice as a holding place for discarded/used flight strips.

IV.C. Simplifications

The realism of the simulator was deliberately decreased during the design and the development of the simulator to account for experimentation rigor and project limitations. At the current moment, only the ground controller position is simulated, mostly to minimize participants needs. No coordination between the apron and local controllers exist (these roles being simulated by pseudos, see Section V), as this type of interaction occurs less frequently on the traffic peaks of interest for this research. Wake turbulence separation



Figure 3. Experimentation on our simulator. Participant face (top to bottom, left to right) out-the-window view, radar screen, flight list and stripboard. Neurophysical data such as eye-tracking, electroencephalography, and electrocardiography are recorded.

was not rigidly enforced - a two-minute departure sequence was modeled but not to the precision as one would need if the local controller role was modeled. Lastly, flights appear and disappear at 5 nm of the airport and fly in straight line final and departure legs. This simplification was determined to be sufficient for a ground controller simulator, but a longer range and SID/Standard Terminal Arrival Route (STAR)-dependent trajectory would likely be needed for the local controller position. The lack of the full 360° out-the-window view does not significantly affect ATCO performance and at minimum, the hotspots of the airport are visible within this field of view. In general, these modifications increase the ability to perform between-participant comparisons and minimize development that was tangential to the specific research needs.

V. Pseudopilots and scenarios

Hardware and software development is difficult for any simulator, but a challenge particular to ATC operations is the simulation of pilot/aircraft interactions with the ATCO. There are at least five elements to the pilot/aircraft interaction with ground controller: 1) the taxiing request from the pilot; 2) taxiing instructions from the ATCO; 3) pilot confirmation of instructions; 4) transfer of frequency from the ATCO; and 5) pilot confirmation of transfer. An ATCO at CDG may receive a new taxiing request every 45-60s (60-80 movements/hr),¹⁷ with each request equaling one aircraft or vehicle. Each pilot may speak one of two languages (English or the native language), have a different accent, or a particular cadence and intonation. While most taxiing instructions given by the ATCO follow standard traffic routes (generally the shortest route), these routes may change based on the ATCO's priorities, the current airport situation, and the workload. Additionally, the ATCO may only give partial clearances, such as a portion of the entire route. The ATCO may ask the pilot yield to another aircraft using the same taxiway, or request that the pilot call back. Subsequently, the controller-pilot interaction is frequent, potentially unpredictable, and has diverse representations.

There are two methods of simulating pilot/aircraft interactions: fully automated or manually via the use of pseudos^h. A fully automated system would need to be capable of adapting to the wide variety of interaction with an ATCO, or require a specific input mannerism that would possibly increase the artificiality of the microworld. Ideally, such an automated system would include voice recognition, spatial interpretation, and oral feedback. The voice recognition aspect would allow recognition of the aircraft callsign when called by the ATCO (such as during the frequency transfer or when priorities are given) and translation of taxiing instructions given by the ATCO. Additionally, such a system would need to be robust to the amount of route details given by the ATCO. For example, ATCOs may provide the main routes and not each individual taxiway, thus requiring the system to find the shortest path to the sub-destination. Spatial interpretation must be included as well, to account for directions relative to the position of the aircraft itself (e.g., "left-right", or "straight ahead"). Lastly, the system must be capable of providing oral feedback, ranging from the basic confirmation of the given taxiing commands to the more complex correction to the taxiing commands given (e.g., if a north departure is routed to a south runway, a pilot would recognize this mistake and correct the ATCO). Such a highly automated system would reduce the need to manually intervene in real-time but will prolong the simulator development time.

The use of pseudos can achieve a relatively high level of operational fidelity, but requires the work of other persons during the execution of a scenario, thus increasing the ratio of researchers to participants from the traditional 1:1 to $n:1$. The simulations runs performed in this simulator required two to three pseudos, depending on the traffic load. Pseudos, if given the proper input devices to the simulator, can adapt to the unpredictability of the ATCO performance. With training, they can react quickly, handle the sparsity of taxiing commands, and recognize non-self-initiated calls. While using pseudos can reduce the amount of simulator development time, it does require additional training and the development of tools to make the appropriate movements and changes in real-time. Given the number of individual events and the expected diversity in participants, we opted to use pseudos and thus created our simulator based on this utility.

There were four main responsibilities of the pseudo: to interpret the taxiing commands to actual trajectories on the airport map; to respond to the ATCO's commands and modify trajectories as necessary; to stop and resume trajectories if in the case of a potential collision; to respect the planned sequence of events in the scenario. The tactile interface provided two input mechanisms: waypoint or tracing. With waypoint, the pseudo could give points along the trajectory and the simulator would find the shortest possible route

^hInteractions with remote adjacent sectors are possible through pseudocontrollers.

between each point. With tracing, the pseudo would manually give the aircraft route on the map. In both cases, the simulator would direct aircraft along these trajectories in accordance to the vehicle dynamics of that particular type of aircraft. The trajectory could be modified, erased, or recalled, with respect to the ATCO's commands. The pseudos monitored their aircraft and those in the vicinity during the scenario. If an aircraft was too close, the pseudo would pause and resume the trajectory as necessary, or call the ATCO for conflict resolution. Lastly, the pseudos attempted to call the tower at the pre-planned frequency but also delayed aircraft with respect to the participant performance. Since these delays also affected other aspects of airport operations, such as the south controllers (apron, local), and north controllers (apron, local, ground), our pseudos were also asked to simulate these responsibilities.

For some airports, certain aircraft types may require more responsibilities than others. Table 2 summarizes which roles are needed for which type of aircraft, with respect to our simulation of the ground controller at CDG. This distribution of additional pseudo work is likely to change if the studied role is not the ground controller or the roles are combined. For each aircraft, the responsible pseudo also simulated the actions of the other ATCO roles. The south local controller handled runway entry and takeoff for the departures, attempting to have a regular sequence. Less emphasis was placed on spacing heavy and medium aircraft. For arrivals, this same controller managed traversing the inner runway, respecting the fact that no aircraft would cross this runway while another aircraft was taking off. The scenario would automatically land an aircraft at given pace. The south apron roles included pushback and parking area taxiing for departures, and parking for arrivals. The specific physical location of the parking stands were only visually simulated; the pseudo would park an aircraft at the closest available parking stand. Two minutes after parking, the aircraft icon disappeared, thus simulating the moment when the pilot would turn off the transponder. Given that the north part of the airport was not easily visually seen by the participant, the local, apron, and ground controllers were not fully simulated. As with the south local controller, the simulator automatically landed arrivals every two minutes. Standard trajectories through the north ground sector were recorded, as to ensure that vehicles would arrive at the most logical taxiway (e.g., F/N or B/Q). As such, the pseudo manually controlled north arrivals or north departures once they reached the vicinity of the south ground sector. In addition for north departures, the pseudo must perform pushback, taxi through the parking area, and north ground taxi the aircraft until the interaction point with the south ground controller.

Table 2. Additional pseudopilot responsibilities based on aircraft origin and destination.

Type of Taxiing	South		North		
	Local	Apron	Local	Apron	Ground
South Departure, parking south	x	x			
South Departure, parking north	x			x	x
South Arrival, parking south	x	x			
South Arrival, parking north	x			x	x
North Departure, parking south		x	x		x
North Arrival, parking south		x	x		x

To ensure the proper sequencing of all these events, scripts were developed for each of the pseudos. These script timelines were built based on the actions described in Table 2 and time approximations for each of the actions. For this research, an estimation of 2-3 minutes was used for taxiing through the parking areas, 7-10 minutes for taxiing through the north end, 2 minutes for pushback, and 10 seconds for providing clearances. These numbers are based on aircraft dynamics and initial simulation validation.

Additionally, the scripts included dialogue with the ATCO, with the proper phraseology, identifiers, and context-specific information for each aircraft. Initial simulator evaluations showed this dialogue significantly aided pseudos who did not have piloting experience, nor were familiar with the airport layout.²⁹ This dialogue also included lingual differences such as number reading (e.g., 874 is read in French as the entire number “eight-hundred-and-seventy-four”, whereas in English the individual digits are split as “eight, seven, four” - both languages may group the digits by two “eight, seventy-four”) and airline identifier (e.g., XLF is referred to as “Starway”, not “X-L-F”). For example, DAH1003, a departure from parking stand B18, would have the several associated actions, as seen in Table 3. With more training, an expert pseudo will be capable of keeping in mind which aircraft are coming up and advancing them as necessary, without a referring to the script timeline. The use of these scripts assured a level of consistency between pseudos and also allowed for a clear division of work. In general, each pseudo handled no more than 15 aircraft. The distribution of aircraft varied and were generally split by a distinguishing factor (to aid in callsign recognition), such as all arrivals, all departures, all aircraft originating or going to the north, or all anglophone aircraft. The work was also distributed based on experience. Novice pseudos generally managed more arrivals than departures, as

arrivals are predictable with respect to initial position and in terms of additional work, require just crossing the inner runway.

Table 3. Examples of pseudopilot scripts based on experience.

Time	Type	Callsign	Location	Activity
07:14:30	Act	DAH1003	B18	Pushback: DAH1003 @ Gate B18 towards G
07:15:00	Call	BEE493K	G30	“Roissy Ground, Jersey 4-9-3 Kilo at Gate Golf-30, requesting pushback”
07:16:50	Act	DAH1003	B18 → G	Apron taxi: DAH1003 towards G
07:17:00	Act	BEE493K	G30 → GW2	Apron taxi: BEE493K towards GW2
07:17:50	Call	BEE493K	GW2 → 27L	“Roissy Ground, Jersey 4-9-3-Kilo on taxiway Golf-Whiskey-2, request taxi.”
07:18:00	Call	DAH1003	G → 26R	“Roissy Ground, Air Algerie 1-0-0-3 on taxiway Golf, requesting taxi.”

(a) Novice: Emphasis on individual actions and phonetic alphabet

Time	Type	Callsign	Location	Activity
07:15:00	Call	BEE493K	G30	“Roissy Ground, BEE493K at Gate G30, requesting pushback”
07:17:50	Call	BEE493K	GW2 → 27L	“Roissy Ground, BEE493K on taxiway GW2, request taxi.”
07:18:00	Call	DAH1003	G → 26R	“Roissy Ground, DAH1003 on taxiway G, requesting taxi.”

(b) Expert: Individual actions and phonetic alphabet are not included.

Our research team consisted of seven pseudos, two of whom held a Private Pilot’s License and five non-pilots. A pseudo training program was developed that consisted of several in-group practice sessions, with one pseudo acting as the ATCO. There are four main competences: phraseology, familiarity with the interface, familiarity with the airport, and pacing. The acquisition of the first competency, phraseology, is aided with the usage of a novice-level script during time in the simulator and does not require simulator time. Familiarity with the interface is gained after more practice inputting trajectories and understanding the capabilities, limits, and reaction times of the input device. Familiarity with the airport layout is not easily achieved without tracing routes in the simulator, but can be aided by studying the airport map and understanding the entry/exit taxiways into each parking area. Lastly, achieving the pacing of the scenario requires the greatest use of resources. The pseudos must react quickly, interpret rapidly dispersed commands, and adhere to the scenario timeline as best as possible. While practicing with actual controllers from the target airport is the ideal situation, one can replicate such fluidity with a non-controller. Since expert controllers are, effectively, reciting standard routes from long-term memory, preparing each aircraft’s standard route to be rapidly read out loud during the practice sessions by the non-controller simulates this effect. Variations can later then be made by stating only the main taxiways, or using directional commands rather than paths. Approximately 8-10 hours of practice time are needed to reach expert levels, but this number may be reduced based on piloting experience. For example, one of our pseudo who was also a pilot achieved competency after a 2 hour practice session.

In addition to the script timelines, definition a simulation scenario consists of generating a textfile that describes the different departures and arrivals along with their flight plans. The most difficult part of this task is acquiring realistic data. All major airports have certain organizational patterns. For instance, Terminals 2E, F, and G at CDG are mainly operated by Air France. Assigning a gate in this terminal to another company would reduce the realism and confuse any professional ATCO working in the scenario. Similarly, it is important to have consistency throughout the scenario definition: the destination airport and SID are closely related, and even if it does not directly impact ground controller work, a discrepancy may distract him or her from the workflow rhythm. The easiest and most accurate solution to these concerns is to use existing training scenarios from the target airport as a starting point for the flight list and the flight plan. Once the scripts are modified as necessary for the research needs, a professional ATCO from the target airport should review and verify the realism.

VI. Feedback

VI.A. Lessons Learned

During the development and execution of this scenario, several lessons were painfully learned. One should carefully consider these points as to avoid delays and other setbacks.

- **A change in configuration results in great variability even in short scenarios.** The scenarios used in this simulator were designed to be 35 minutes long (due to ATCO availability), with the QFU change warning declared T+5 minutes from the start of the scenario and the last arrival in the old configuration landing at T+15. Even in 10 minutes, the departure plane situation was very different from one ATCO to another, leading to more or less (and not the same) aircraft to be rerouted to the other end of the runway. This situation complicates the between-participant comparison capacity.
- **Pre-recording some traffic (both as simulation input/bootstrap, or to reduce the load) for the pseudos can lead to additional complications.** While useful in many cases, pre-recorded traffic can add another demanding monitoring task to pilots and lowers the situation awareness compared to monitoring pilot-directed aircraft. The situation can be so variable from one simulation to another that even recording pushback or apron movement was not possible, as it would result in a traffic jam in the apron area.
- **Synchronised pseudo interfaces increases situation awareness and allows silent collaboration.** Displaying the same information on each of the individual pseudo interfaces improves coordination. Each moving order trajectory feedback was displayed on all pseudos screen for a few seconds, this proved efficient for silent collaboration, for instance, when a pseudo send an aircraft to takeoff, the others are aware that crossing the inner runway is forbidden for the next 30 seconds.
- **Precise timing is impossible to achieve in heavy traffic scenarios.** There is too much deviation that occurs due to individual ATCO performances. For this research, the sequence of events was maintained more so than the exact timing. This choice is close to reality with aircraft being retained on their stand to minimize aircraft movement around the airport. In times of heavy traffic, especially with the use of current interface technology and radio communication, the targeted time of a call to the tower by a pseudo can be delayed up to as much as a minute for a free radio slot.

VI.B. Validation

In addition to feedback and discussion with actual ATCOs, we aimed to formally quantify the validity of the simulator and to allow for comparison to other simulators of CDG. To the authors' knowledge, no formal simulation validation methodology exists. However, Feinstein and Cannon³⁰ developed a two-part framework of simulation validation based on the degrees of representation and education. As this simulator was not intended for educational purposes, which, as noted in their literature review may necessitate lower fidelity simulators, this research focused primarily on representational validation. We developed a questionnaire based on the constructs assembled by the authors, covering simulation verification and seven dimensions of validity: conceptual, construct, content, convergent, criterion, external, and internal. There was a statement for each of the eight constructs (evaluated on a 5-point Likert scale) and two open-ended questions regarding the elements that were useful or missing or unnecessary for the simulator realism. The final score was the average of each dimensions rating. Table 4 presents all of the questions.

In the case of this project, the project goals were new operational concepts; the studied task was the ground controller at a large airport; and actual working conditions are at a large airport similar to CDG. The questions were translated into French. We added an additional question regarding external validity. Our research group was also interested in the extrapolation of the results at an airport such as CDG to other airports of other sizes and origins. As such, we posed two versions of question 7 (differing aspects bolded for clarity):

- a The Simulator is realistic enough that results and trends from the User performance of the ground controller conducted in the Simulator could be extrapolated to real world performance at Roissy Charles-de-Gaulle.

Table 4. Simulator Evaluation Questionnaire

No.	Construct	Question
1	Verification	The Simulator is realistic enough to evaluate the project goals.
2	Conceptual Validity	The Simulator accurately portrays the physical and operational environments of <u>the studied task</u> .
3	Construct Validity	The Simulator <u>correctly</u> presents the relationships between the different aspects of the environment of <u>the studied task</u> in actual working conditions, such that changing one aspect would affect the other aspects as expected.
4	Content Validity	The Simulator is detailed and thorough in its representation of the different elements (physical and operational) for <u>the project goals</u> .
5	Convergent Validity	If faced with the exact same scenario in <u>the exact same role</u> in the real world, my two performances would not difference, since I would have applied the same level of risk or precaution in the Simulator.
6	Criterion Validity	The Simulator accurately responds and reflects the experience and the skill levels of the User (someone more/less skilled or experienced than me would perform better/worse).
7	External Validity	The Simulator is realistic enough that results and trends from the User performance of <u>the studied task</u> conducted in the Simulator could be extrapolated to real world performances in <u>actual working conditions</u> .
8	Internal Validity	The Simulator correctly represents <u>plausible sources of the events during the studied task and the actual working conditions</u> , the logic is representative of <u>actual working conditions</u> .

- b The Simulator is realistic enough **and the representative situations are sufficiently comprehensive** such that the results and trends from the User performance of the ground controller conducted in the Simulator could give an estimation of the real-world performance **of new technology concepts at any airport**.

This questionnaire was distributed *post-hoc* to the ATCOs who had participated in a prior experiment. They were asked to evaluate two simulators: ours and the training simulator at ENAC. Controllers from CDG were specifically asked to rate the simulator at CDG, as it is the only airport that has its own training simulator. Six participants took part in this survey, with six reviews of our simulator, three of CDG, and two of ENAC, and one of Orly airport. The final score was calculated by averaging the individual scores of the eight questions.

On average, the six participants rated our simulator 3.46 out of a maximum of 5 points ($\sigma = 0.74$, min score = 2.75, max score = 4.88). Participants generally appreciated the physical work environment and the realism of aircraft movements. The external view was well appreciated by some, while others noted that the entire 360° view was not fully modeled. Participants also wished for a larger or additional strip board. Participants stated that the traffic load was realistic and the aircraft reacted quickly to the participant's commands. However, for some participants, the aircraft did not react quickly enough, thus reducing the participant's ability to predict upcoming traffic movements. Participants did not like the fact that the other controllers were not simulated and the teamwork between each of the roles was not explored, even though that was not the point of our research.

In general, participants felt that our simulator was realistic enough to evaluate the project goals ($\mu_{\text{valid}} = 4.33$, $\sigma_{\text{valid}} = 0.52$) and accurately reflected user's experience and skill ($\mu_{\text{crit}} = 4.00$, $\sigma_{\text{crit}} = 0.63$) but that it lacked conceptual validity ($\mu_{\text{concept}} = 3.00$, $\sigma_{\text{concept}} = 1.41$). Several participants noted that if approached with the same scenario in the real-world, they would not react the same way ($\mu_{\text{conv}} = 3.00$, $\sigma_{\text{conv}} = 1.67$). Overall, the participants were slightly in agreement on the applicability of the results to both CDG and other airports, with questions 7a and 7b scoring an average of 3.17 ($\sigma_{7a} = 1.17$) and 3.33 ($\sigma_{7b} = 0.82$) respectively. This feedback provides reassurance regarding our choice of using CDG as the paradigm for testing new automated taxiing technologies.

As a comparison point, the participants were also asked to rate the simulator of their home airport. On average, the three participants who rated the CDG simulator gave it a score of 4.75 of 5 points ($\sigma = 0.125$, min score = 4.63, max score = 4.88). They noted that simulator was well-adapted to the actual work environment (albeit with changes, such as the use of a mouse for the radar screen, or non-simulated buildings) and allowed for multiple controllers and roles, including interactions with other members of the team such as the Tower

Supervisor, firefighters, the ATCOs handling final approach, etc. This subset of participants rated our simulator, on average, 4.04 points. Two participants provided ratings for the simulator at ENAC, which scored an average of 3.44 points (min score = 3, max score = 3.875). These same participants rated our simulator 3.31 points. The response for Orly was ignored due to the singular data point.

VII. Conclusion

The work involved with developing an air traffic control tower simulator is modular, both in terms of different components and capacity. Different aspects can be completed in parallel or improved upon over time. Depending on the capacity of the simulator and the complexity of the scenario, it may require at least one other person (acting as the pseudopilot) for its operation, in addition to the participant. Subsequently, such a project is feasible at all research levels, including university research labs. We developed and validated a medium-high simulator of the south end of Roissy Charles-de-Gaulle in Paris, France for the ground controller that can simulate up to 75 movements/hr, using between 1-3 pseudopilots. To support this work, a list of most important simulation elements, pseudopilot scripting materials, and a formal simulation validation questionnaire were also created. In addition to its clear usage as a training platform and means of testing technology concepts, such a simulator may be used to examine the extremes of air traffic controller performance (monotony due to low workload, stress and performance limits due to high workload), forecasted workloads (nonexistent in current operations), and changes in teamwork with the inclusion of automation. In terms of this particular research, the simulator has been validated to be sufficiently useful for evaluating ideas and concepts and to ensure that the introduction of such technology is not detrimental to the workload of the air traffic control officer.

Acknowledgments

The authors thank François Lancelot and the rest of the Modern Taxiing team for their continued hard work. We thank Raïlane Benhacène, Géraud Granger, Michael Traoré and Brice Beldjilali for volunteering as pseudopilots and the anonymous controllers who participated in this project. This work is co-financed by EUROCONTROL acting on behalf of the SESAR Joint Undertaking (the SJU) and the EUROPEAN UNION as part of Work Package E in the SESAR Programme. Opinions expressed in this work reflect the authors' views only and EUROCONTROL and/or the SJU shall not be considered liable for them or for any use that may be made of the information contained herein.

References

- ¹Jones, T. and Takemoto, P., "Fact Sheet - Tower Simulation System (TSS)," [Online]. Available: http://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=10156.
- ²Jones, D. G. and Endsley, M. R., "Overcoming representational errors in complex environments," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 42, No. 3, 2000, pp. 367–378.
- ³NASA Ames Research Center, "NASA Ames Aviation System Division: FFC Main," [Online]. Available: <http://www.aviationsystemsdivision.arc.nasa.gov/facilities/ffc/>.
- ⁴Dorigi, N. and Sullivan, B., "FutureFlight Central: A Revolutionary Air Traffic Control Tower Simulation Facility," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, American Institute of Aeronautics and Astronautics.
- ⁵Verma, S., Kozon, T., Lozito, S., Martin, L., Ballinger, D., and Cheng, V., "Human Factors of Precision Taxiing under two levels of Automation," *Air Traffic Control Quarterly*, Vol. 18, No. 2, 2010, pp. 113–141.
- ⁶The MITRE Corporation, "Integration Demonstration and Experimentation for Aeronautics Laboratory," Online, 2012.
- ⁷Stelzer, E., Stanley, R. M., and Shepley, K. K., "Evaluating Surface Trajectory-Based Operations concepts through a human-in-the-loop simulation," *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, Oct 2011, pp. 3D2–1–3D2–12.
- ⁸Thales, "ScanSim: A comprehensive and cost-effective ATC simulator suite for ATC training," Online, Rungis, France, July 2013.
- ⁹EUROCONTROL, "Real Time Simulations," [Online]. Available: <https://www.eurocontrol.int/articles/real-time-simulations>.
- ¹⁰Martin, L., Swenson, H., Thipphavong, J., Chen, L., and Seo, A. Y., "Effects of Scheduling and Spacing Tools on Controllers' Performance and Perceptions of their Workload," *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, 2010.
- ¹¹Prevot, T., "Exploring the many perspectives of distributed air traffic management: The Multi Aircraft Control System MACS," *Proceedings of the HCI-Aero*, 2002, pp. 149–154.

- ¹²Jakobi, J., Teotino, D., and Montebello, P., "Towards Higher-Level Service of an Advanced Surface Movement Guidance and Control System (A-SMGCS)," *Air Traffic Control Quarterly*, 2010.
- ¹³Straussberger, S. and Schaefer, D., "Monotony in air traffic control," *Air traffic control quarterly*, Vol. 15, No. 3, 2007, pp. 183–207.
- ¹⁴Small, R. L. and Hammer, J. M., "Comparing Display Symbolology for an Advanced Air Traffic Control Tower Application," *IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans*, 1997.
- ¹⁵Fothergill, S., Loft, S., and Neal, A., "ATC-labAdvanced: An air traffic control simulator with realism and control," *Behavior research methods*, Vol. 41, No. 1, 2009, pp. 118–127.
- ¹⁶Chua, Z., Causse, M., Cousy, M., and Andre, F., "Modulating Workload for Air Traffic Controllers during Airport Ground Operations," *Human Factors and Ergonomics Society Annual Meeting*, Human Factors and Ergonomics Society, October 2015.
- ¹⁷Directorate General for Civil Aviation, *Manuel d'Exploitation TWR/APP*, Direction des Services de la navigation aeri-
enne, ed: DGAC/DSNA/DO/SNA-RP/CDG/SE ed., 2008-2009.
- ¹⁸UFA, "ATTower," [Online]. Available: <http://www.ufainc.com/products/attowertowsim-tower-simulation/>.
- ¹⁹Peel, R., "XPlane Data," [Online]. Available: <http://data.x-plane.com/>.
- ²⁰OpenStreetMap Foundation, "OpenStreeMap," [Online]. Available: <http://www.openstreetmap.org/>.
- ²¹Haklay, M., "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets," *Environment and Planning B Planning and Design*, , No. 37, 2010, pp. 682–703.
- ²²Schweikert, F., "[Talk-at] Flughafen Wien - VIE," [Online]. Available: <https://lists.openstreetmap.org/pipermail/talk-at/2013-September/005914.html>.
- ²³Pavet, D., "Use of Paper Strips by Tower Air Traffic Controoler and Promises Offered by New Design Technique on User Interface," *Proceedings of the 4th ATM Seminar*, Santa-Fe, 2001.
- ²⁴Olson, C., "FlightGear Flight Simulator," [Online]. Available: <http://www.flightgear.org>, Open-source software.
- ²⁵Chatty, S., *The ivy software bus*, Centre d'Etudes de la Navigation Aerienne, Toulouse, France, NT02-816 ed., 2006.
- ²⁶Buisson, M., Bustico, A., Chatty, S., Colin, F.-R., Jestin, Y., Maury, S., Mertz, C., and Truillet, P., "Ivy: Un Bus Logiciel Au Service Du Developpement De Prototypes De Systemes Interactifs," *Pr. of the 14th French-speaking Conference on Human-computer Interaction*, IHM '02, ACM, New York, NY, USA, 2002, pp. 223–226.
- ²⁷TeamSpeak Systems GmbH, "TeamSpeak," [Online]. Available: <http://www.teamspeak.com/>.
- ²⁸"Mumble," [Online]. Available: <http://www.mumble.info/>.
- ²⁹Chua, Z. K., Cousy, M., Andre, F., and Causse, M., "Simulating Air Traffic Control Ground Operations: Preliminary Results from Project Modern Taxiing," *4th Annual SESAR Innovation Days*, Single European Sky ATM Research, Madrid, Spain, 2014.
- ³⁰Feinstein, A. H. and Cannon, H. M., "Constructs of simulation evaluation," *Simulation & Gaming*, Vol. 33, No. 4, 2002, pp. 425–440.