AMAN Case Study

Philippe Palanque[1][0000−0002−5381−971X] and
José Creissac Campos[2,3][0000−0001−9163−580X]

[1] Interactive Critical Systems Group, Université Toulouse III Paul Sabatier
palanque@irit.fr
[2] Department of Informatics, University of Minho
[3] HASLab, INESC TEC
jose.campos@di.uminho.pt

Abstract. This document presents the case study for ABZ 2023 conference. The case study introduces a safety critical interactive system called AMAN (Arrival MANager) which is a partly-autonomous scheduler of landing sequences of aircraft in airports. This interactive systems interleaves Air Traffic Controllers activities with automation in AMAN. While some AMAN systems are currently deployed in airports, we consider here only a subset of functions which represent a challenge in modelling and verification.

Keywords: Interactive systems · formal methods · Case study · Automation · AMAN · Air Traffic Control.

Version history

Version	Date	Comment
1.0	16/09/2022	First version of document.

1 Introduction

The Air Traffic Control activity in the TMA (Terminal Manoeuvring Area) is an intense collaborative activity involving at minimum two air traffic controllers working in a shared workspace (see image below) communicating with a set of aircraft. The TMA is the area where controlled flights approach and depart in the airspace close to the airport.

Air Traffic Control (ATC) is a collaborative work performed locally by two specialised air traffic controllers. The executive (EXEC) Air Traffic Controller (ATCo) interacts with pilots (usually using voice) while the planner (PLAN) ATCo organises the work and the flow of aircraft in the area.

The planner controller (left-hand side of Figure 1) is in charge of organising and planning the traffic. This could result in changing the aircraft flight plan such as heading, speed, altitude. Requests for such changes are given by EXEC ATCo (usually using voice) who uses a radar screen (see right-hand side of Figure 1). The EXEC ATCo is the controller deputed to handle the communications ground/air/ground, communicating to the pilots and releasing clearances

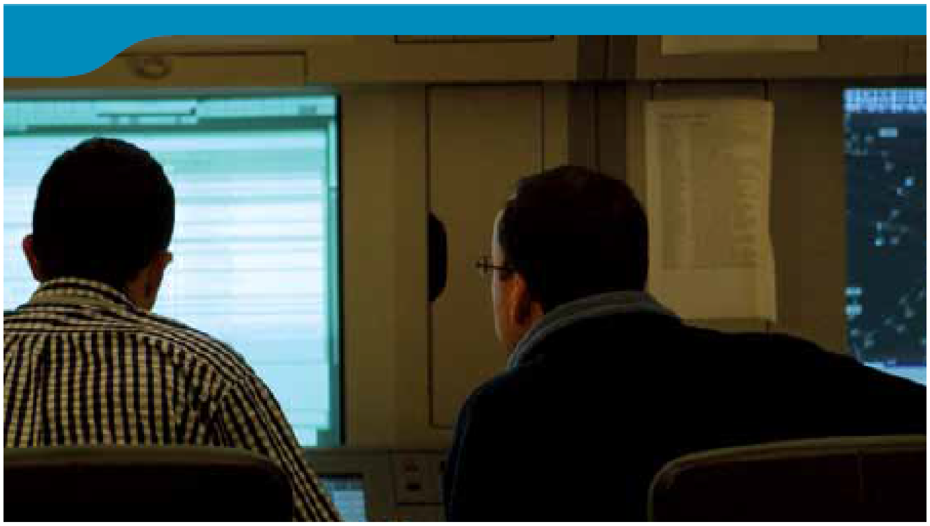


Fig. 1. Executive and planner ATCs

to aircraft. He/she has the tactical responsibility of the operations and he/she executes the AMAN advisories to sequence aircraft according to the sequence list.

For the case study scenario, we propose that the pilots assume a passive role, limited to the reception and execution of the clearances. Other more active roles (such as requesting an emergency landing) can be considered but are likely to make things significantly more complex.

Thus, the case study will focus on a subpart of the work that consists in organising the sequencing of landing of the aircraft on the runway(s).

2 Overview of the AMAN tool

The AMAN (Arrival MANager) tool is a software planning tool suggesting to the PLAN ATCo an arrival sequence of aircraft targeting at providing support in establishing the optimal aircraft approach routes. Its main aims are to assist the controller to optimize the runway capacity (land as many aircraft as possible and as quickly as possible) and/or to regulate/manage (meter) the flow of aircraft entering the airspace, such as a TMA [5]. It helps to achieve more precisely defined flight profile and to manage traffic flows, in order to minimize the airborne delay, leading to better efficiency in terms of flights management, fuel consumption, time, and runway capacity utilization.

The AMAN tool uses the flight plan data, the radar data, an aircraft performance model, known airspace/flight constraints and weather information to provide to the traffic controllers, via electronic display, two kind of information:

- A Sequence List (SEQ_LIST) which is an arrival sequence that optimizes the efficiency of trajectories and runway throughput (see Figure 2)
- Delay management Advisories which presents the delay (with respect to flight plan) for each aircraft in the ATCo's airspace.

Figure 2 presents an abstract view of AMAN tool showing (by means of arrows) the workflow of the tool that:

- exploits flight plan information, radar and weather information (left-hand side of the figure);
- performs predictions about the arrival time of the aircraft on the runway
- exploits safety spacing requirements and the predictions to compute a landing sequence that will be presented to the PLAN ATCo and may be used by that person.

In this part of the description of the tool we consider only AMAN as an information presentation tool. Later we will present some requirement for an interactive tool meaning that the proposed landing sequence may be tuned by the PLAN ATCo. At the bottom right-hand side of Figure 2, we can see that AMAN (according to Eurocontrol specifications in [5] page 3) may also produce and present a list of advisories which may be sent to aircraft pilots by EXEC ATCo in the form of clearances requesting a modification of speed to meet the computed schedule. In the rest of this case study description we will not take into account this part of the functioning of AMAN, and will instead assume that the ATCos will identify the required clearances from the information displayed.

Each of the next sections will cover one aspect of the tool, from Prediction to the tasks of the ATC.

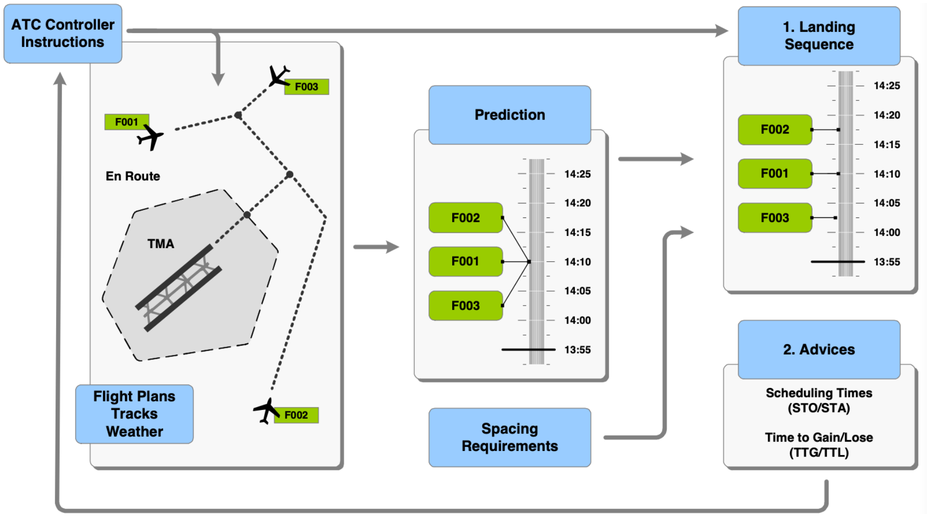


Fig. 2. High-level view of the AMAN tool

2.1 Prediction

Each aircraft is following a flight plan containing (among others) the aircraft type (which is useful for knowing maximum speed), the expected flight time, arrival airport, departure flight time, flight time autonomy. The prediction part of AMAN merges the information available in the registered flight plan with real-time information provided by radars and predicts an arrival time for each incoming aircraft.

2.2 Spacing requirements and computation of a landing sequence

According to international regulations, the work of ATCos is to ensure flight safety by keeping vertical and horizontal separations between aircraft in a sector. In the higher airspace such separation is 5 NM (nautical miles horizontally) and 1000 ft (feet vertically) between each aircraft. When entering the TMA, this separation is not maintained anymore but (to avoid incidents and accidents due to turbulence and to provide enough time to react in case of problem) a landing separation of 3 minutes between aircraft is requested. Except under exceptional circumstances this 3 minutes separation must be ensured by the ATCos and by AMAN. Depending on the number of aircraft on arrival, this might be a complex constraint of which the satisfaction may require speeding up or slowing down some aircraft but also having some aircraft on HOLD which means sending them to a waiting zone for later processing. In such a case, the aircraft will be removed (after a while) from the landing sequence.

2.3 AMAN User Interface

Figure 3 is an example of a concrete AMAN user interface. It could be relevant to define and represent interactions from controllers such as using drag and drop interaction technique to modify the sequencing proposed by AMAN prediction tool.

As one can see, this user interface is rather complex with display of a lot of information relevant to the various facets of the work of TMA ATCos. For the case study we will propose a simplified but realistic user interface (see Figure 6) focusing on a subset of critical tasks in relation with the use of the AMAN tool.

2.4 Air-Traffic Controller Tasks

Certification Specification CS 25 [2] paragraph 1302 states that ”This paragraph applies to installed equipment intended for flight-crew members’ use in the operation of the aeroplane from their normally seated positions on the flight deck. This installed equipment must be shown, individually and in combination with other such equipment, to be designed so that qualified flight-crew members trained in its use can *safely perform their tasks* associated with its intended function ...”. Acceptable means for compliance to meet such requirement would require to explicitly and exhaustively describe operators’ tasks.

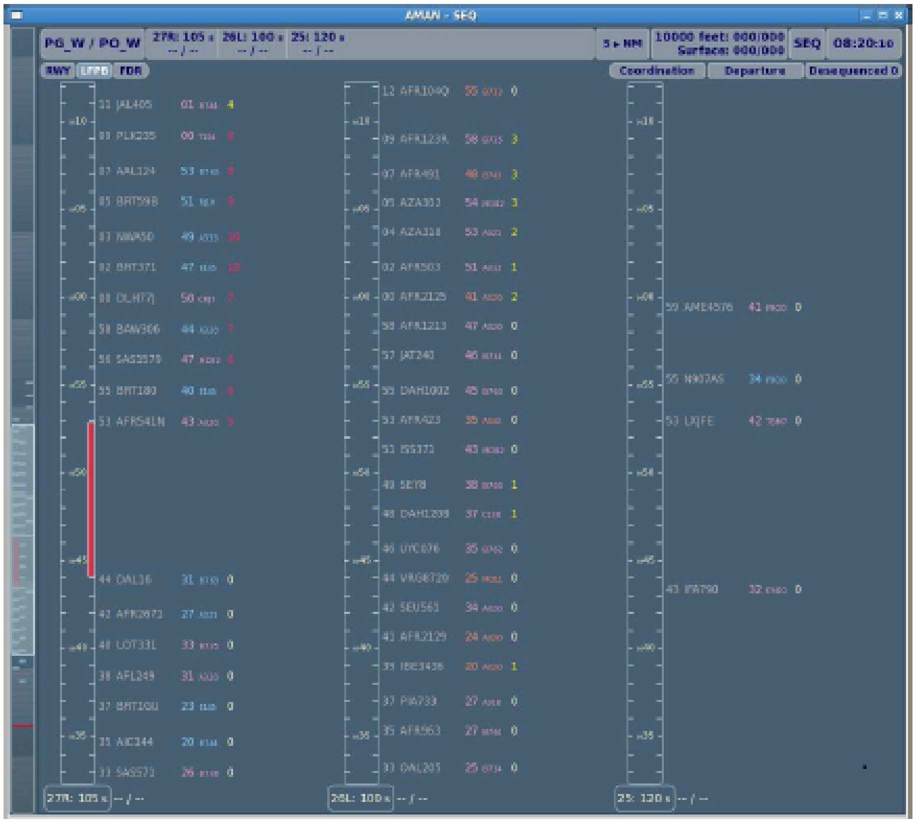
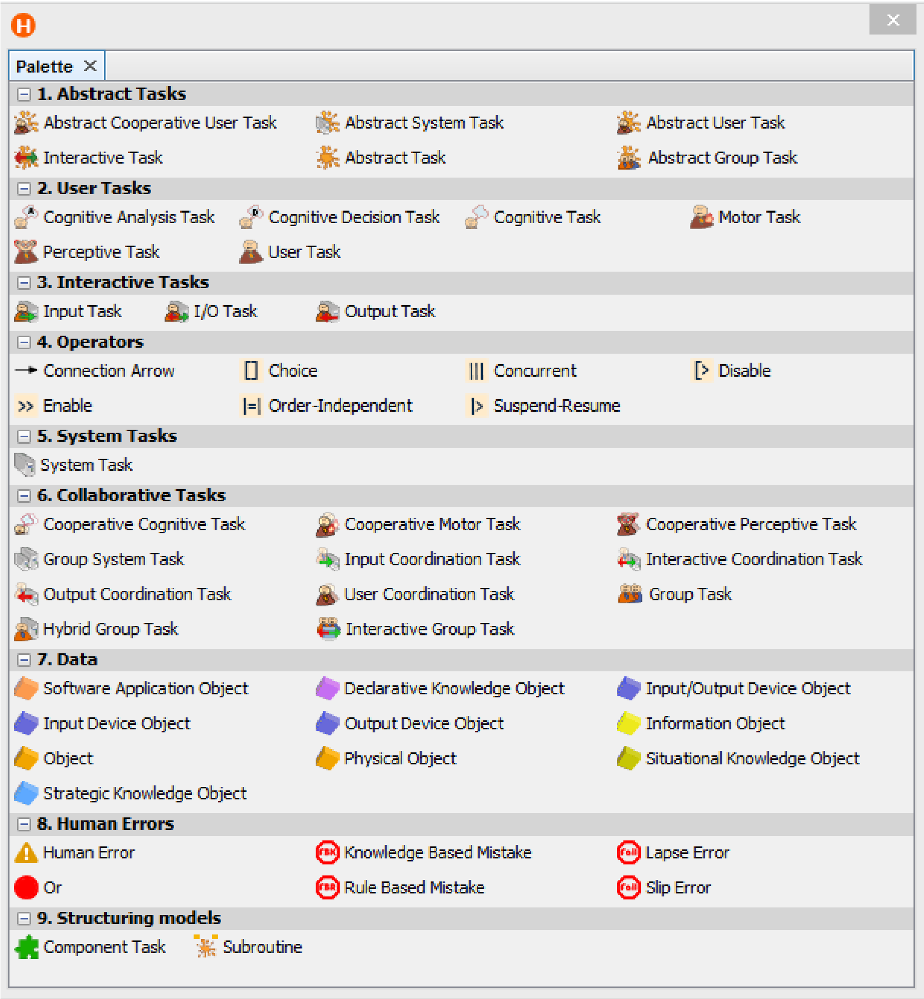


Fig. 3. MAESTRO AMAN tool UI example from [1]

HAMSTERS (Human – centered Assessment and Modelling to Support Task Engineering for Resilient Systems) is a tool-supported task modelling notation for representing human activities in a hierarchical and temporally ordered way [9]. The HAMSTERS notation provides support for representing a task model, which is a tree of nodes that can be tasks or temporal operators. The top node represents the main goal of the user, and lower levels represent sub-goals, tasks and finally actions. Task types are elements of notation that enable to refine and represent the nature of the task as well as whether it is the user or the system who performs the task. The main task types are abstract, user, interactive and system tasks. HAMSTERS tool makes it possible to refine such tasks to describe more precisely operator's actions such as representing motor, perceptive and cognitive tasks involved in the accomplishment of a goal.

Abstract tasks (part numbered 1 in Figure 4) provide support to describe sub-goals in the task model. They also provide support to describe tasks for which the refinement is not yet identified, at the beginning of the analysis process. User tasks (part numbered 2 in 4) provide support to describe the detailed



Palette ✕

⊟ **1. Abstract Tasks**

Abstract Cooperative User Task · Abstract System Task · Abstract User Task

Interactive Task · Abstract Task · Abstract Group Task

⊟ **2. User Tasks**

Cognitive Analysis Task · Cognitive Decision Task · Cognitive Task · Motor Task

Perceptive Task · User Task

⊟ **3. Interactive Tasks**

Input Task · I/O Task · Output Task

⊟ **4. Operators**

→ Connection Arrow · [] Choice · ||| Concurrent · [> Disable

>> Enable · |=| Order-Independent · |> Suspend-Resume

⊟ **5. System Tasks**

System Task

⊟ **6. Collaborative Tasks**

Cooperative Cognitive Task · Cooperative Motor Task · Cooperative Perceptive Task

Group System Task · Input Coordination Task · Interactive Coordination Task

Output Coordination Task · User Coordination Task · Group Task

Hybrid Group Task · Interactive Group Task

⊟ **7. Data**

Software Application Object · Declarative Knowledge Object · Input/Output Device Object

Input Device Object · Output Device Object · Information Object

Object · Physical Object · Situational Knowledge Object

Strategic Knowledge Object

⊟ **8. Human Errors**

Human Error · Knowledge Based Mistake · Lapse Error

Or · Rule Based Mistake · Slip Error

⊟ **9. Structuring models**

Component Task · Subroutine

Fig. 4. Tool palette in HAMSTERS

human aspects of the user activities. User task types can be refined into perceptive, motor, cognitive analysis, and cognitive decision tasks. For example, the user may perform a motor task (such as grabbing a card) or cognitive task (such as remembering a PIN code). Such refinement enables the analysis of several aspects of the tasks performed by the user such as cognitive load, motor load, required perceptive capabilities. Such refinement also enables to identify possible threats that can be associated to specific types of user actions. Temporal operators are used to represent temporal relationships between sub-goals and between activities. Interactive tasks (part numbered 3 in Figure 4) provide support to describe tasks that are action performed by the user to input information to the system (interactive input task) or action perform by the system to provide information to the user and that are meant to be perceived by the user (interactive output task). Interactive input/output tasks provide supports to describe both cases. System tasks (part numbered 5 in Figure 4) provide support to describe the tasks that the system executes. The system may execute an input task, i.e. the production and processing of an event produced by an action performed by the user on an input device. It may also execute and output task, i.e. a rendering on an output device (such as displaying a new frame on a screen). The system may execute a processing task (such as checking the user login and password). In addition to elements of notation for representing user activities and their temporal ordering, HAMSTERS provides support to represent data (e.g. information such as perceived amount of money on an account, knowledge such as a known password), objects (e.g. physical objects such as a credit card, software objects such as an entered password using a keyboard) and devices (e.g. input devices such as keyboard and output device such as a screen) that are required to accomplish these activities (part numbered 7 in Figure 4). HAMSTERS and its eponymous interactive modelling environment is the only environment providing structuring mechanisms as real-life models are usually large and reuse is useful [9].

Tasks of the EXEC ATCo are described using the HAMSTERS notation [9] and [4] (see Figure 5). The notation presented in [7] explicitly supports collaborative activities among users but this is not exploited here as we focus only on the work of the PLAN ATCo. This notation can also be used not only to describe nominal activities of operators but also the errors they may perform and the activities necessary to recover from them [3].

Figure 5 should be read from top to bottom and from left to right. When LS appears in the model, it is an acronym for "Landing Sequence". The top of the image describes the main goal of the operator which is to manage the TMA sector. This activity consists in two tasks, manage the landing sequence called "Manage LS" on the figure and "Stop Manage LS" which deactivates (operator |>) the repetitive task "Manage LS" (see loop symbol on the left-hand side of the icon of the task) and terminates the task. The "Manage LS" task is decomposed into two sub-tasks. The first one called "Manage Landing Sequence (LS)". this task is interrupted every 10 seconds by the autonomous behaviour of AMAN. This is represented by the abstract system task "AMAN Autonomous

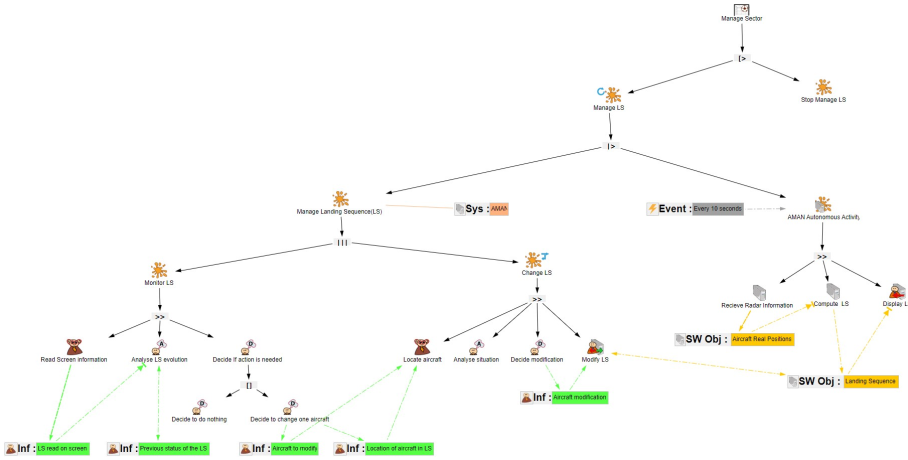


Fig. 5. PLAN ATCo tasks in HAMSTERS (zoom in for details)

activity" which is performed every 10 seconds. This task is decomposed into three tasks which are performed by the tool in sequence (operator >> in the model): "Receive radar Information", "Compute LS" and "Display LS". This task is an output system task (icon with a red arrow) meaning that the task will change the display (to be read by the ATCo). For these tasks, two software objects are used: the "Aircrat Real Positions" provided by the radar and used by AMAN by "Compute LS" task which produces the software object "Landing Sequence".

3 The landing sequence User Interface

As explained above, we propose here a simplified user interface of an AMAN tool. For instance, we don't take into account the production of advisories that would support the ATCos in identifying the clearances to be send to pilots. In this section we describe in detail this simplified user interface (see Figure 6). Next section focuses on the graphical appearance of the user interface. The following section details the interaction techniques that are used by the ATCos to provide input to AMAN. Last section refines the task model presented in 5 taking into account the user interface and interactions.

3.1 AMAN Simplified User Interface

The user interface presents a graphical representation of the AMAN advisory horizon on the left. This includes the current time at the bottom (in this case, 18:02) and, above it, a timeline against which flight labels are positioned.

Flights labels point to their Predicted Time of Arrival at the runway (predicted by AMAN). Each Label identifies the flight number and the arrival time

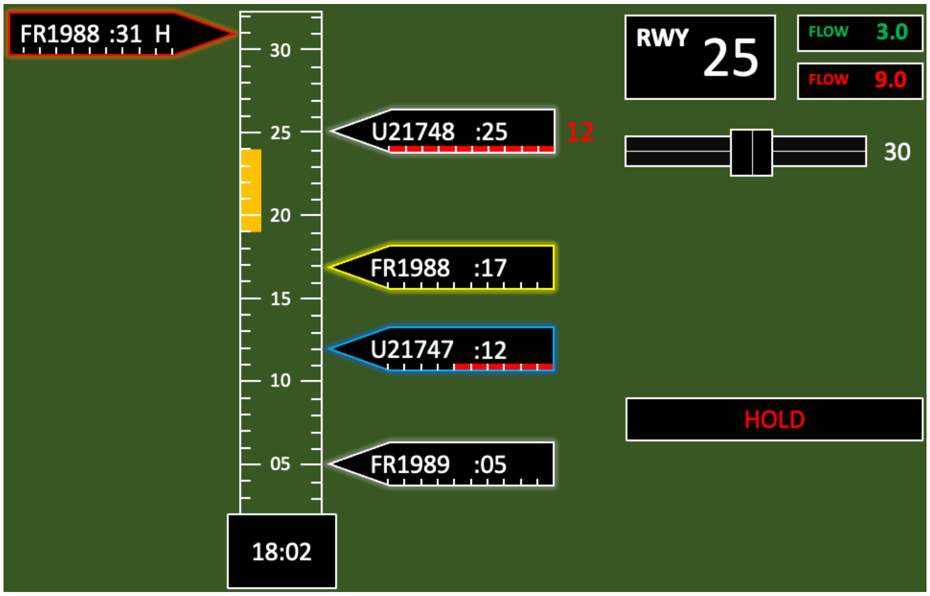


Fig. 6. Idealised AMAN landing sequence UI

(minutes). If the flight needs to absorb a delay to keep to the assigned landing time[4], that is indicated by a red bar at the bottom of the strip. Delays of up to 10 minutes are represented by the bar's length (with each tick representing one minute to absorb). For longer delays (when the bar is full), the number of minutes to absorb is indicated in red next to the label (see flight UL21748, which has a delay of 12 minutes). Negative values can also be indicated (in green) and represent situations where the plane needs to speedup to meet the assigned time).

If a flight is on hold, that is indicated by an ”H” in the label. This is a temporary display as the flight will be removed from the landing sequence and will reappear at a later stage (when called for landing by the EXEC ATCo).

The flight label (line) is colour coded to indicate the flight status: “*yellow for flights that are unstable (the order of the flight in the sequence and its runway current allocation may change), blue for the flights that are stable (the order of the flight in the sequence may change while its runway allocation is definitive), and white for the flights that are “freezed” (the order of the flight in the sequence and its runway allocation are definitive)´´* [6]. When on hold, the aircraft label is coloured in red until it is removed by AMAN from the landing list. The timing information about the landing sequence is presented in white with the actual time displayed at the bottom, as already mentioned above.

For different reasons, such as runway cleaning or when ground vehicles are in operation on it, periods of time can be blocked by the PLAN ATCo, in which case

[4] I.e., the flight is early in relation to the assigned landing time.

they are marked in yellow. Such a locked period is visible on Figure 6 between 19 and 24 minutes. This means that AMAN will not position any landing in that slot.

On the right side of the interface in Figure 6 there is information on the runway, the status of traffic, and some commands to be used by the ATCO. At the top, the runway is indicated, as well as the flow information on the two runways (for simplicity we focus in this case study on one runway only). The flow represents the number of aircraft currently present in the landing sequence. On this example we see that there are 9 aircraft but only 5 are displayed on the UI. This is due to the fact that there is a level of zoom that is currently hiding 4 aircraft. If the flow is green, additional capacity is available. This information is useful to the ATCo for instance for removing aricraft on hold. If the flow is red, the runway is overloaded and it is not recommended to add more aircraft to it. Below this information, there is a slide-bar to change the zoom level. which determines how much in the future the horizon extends. The user interface only shows those flights that fall inside the current zoom level. In this case the zoom level is set to show 30 minutes into the future and thus only displays aircraft labels that are predicted to landing within the next 30 minutes.

Finally, the button labelled HOLD allows the PLAN ATCo to inform AMAN to "remove" aircraft from the list.

3.2 AMAN Interaction

Interaction on the timeline Interaction on the timeline is limited to changing an aircraft label by moving it up and down. If the target position is already partially used by another aircraft label, the moving aircraft level will be moved on the other side of the timeline (left or right). In order to keep the three minutes separation for every aircraft in the landing sequence, the aircraft label must have three empty spaces with the other labels.

Figure 7 presents the graphical appearance of the direct manipulation of an aircraft label. First the aircraft label FR1989 is selected by positioning the mouse cursor over the label and by pressing the left mouse button. Keeping the mouse button and moving the mouse will dynamically instantiate a new graphical object (usually called a ghost) with the same information as the aircraft label but with graphical attributes with a level of transparency of 50 percent. This graphical object can be moved up and down but remains snapped to the timeline (it is not possible to move it left or right). When the mouse button is released, the ghost aircraft label graphical attributes are set to 0 percent transparency. This aircraft label is positioned in front of the closest dash on the timeline. The aircraft label at the original position is deleted. for safety reasons it is important to guarantee that the aircraft labels (not taking into account the ghosts) do not overlap.

Interaction on the Zoom slider The zoom slider is on the right-hand side of the user interface. The current zoom value is displayed next to the slider (currently the zoom value is 30). The zoom value can move from 15 minutes to

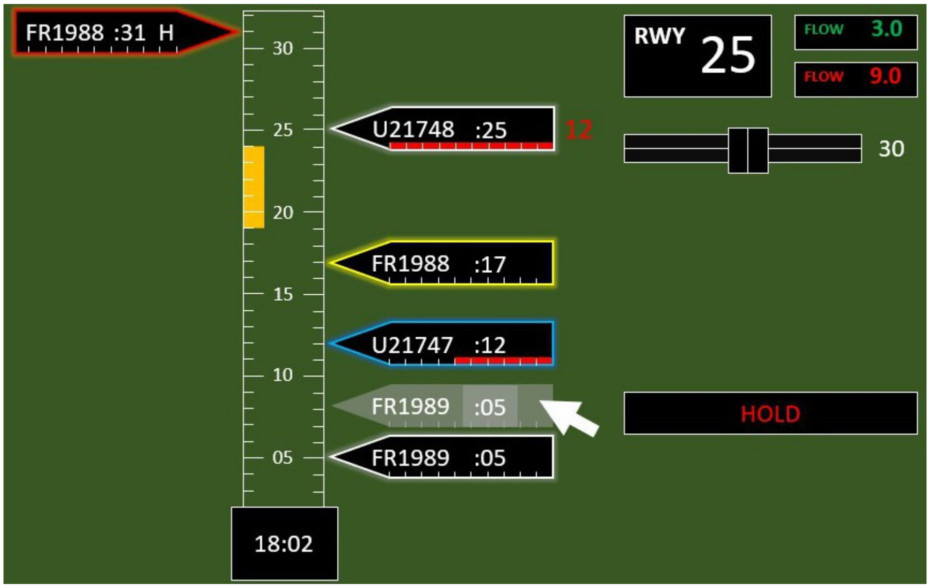


Fig. 7. Direct manipulation of aircraft label

45 minutes. On the slider, the current position is represented by the lift (black square). The lift can be directly manipulated with the mouse by moving the mouse cursor on the black square, pressing the left button, moving to the right (to increase the value) or to the left to decrease the value manipulating the lift. The value of the zoom moves by jumps of 5 minutes meaning that the acceptable values are 15, 20, 25, 30, 35, 40 and 45 only. It is thus necessary to move the mouse cursor for more than 0.5 cm to move to the next acceptable value. If the mouse cursor is moved beyond the slider limits (left or right) the movements have no effect on the selected value. When the mouse button is released, the display is updated showing all the aircraft labels in the landing sequence which will be landing in less than ”zoom value” minutes. The other aircraft (if any are not displayed).

Interaction on the HOLD button The HOLD button behaves as a standard button. For the function associated to the button to be triggered, a flight must have been previously selected with the mouse, the mouse cursor must be positioned on the HOLD button, the left mouse button pressed and released (on the mouse button). If the mouse button is released outside of the HOLD button, the action is not trigger. When the mouse button is pressed on the HOLD button, the graphical appearance of the button is changed (as shown in 8).

Blocking a time slot It is possible for the user to block a time slot on the timeline (as seen in yellow in 6 between 19 and 25 seconds. In order to add a new blocked time slot, the mouse cursor must be positioned on the left-hand side of the timeline. Clicking with the mouse at a given position will add a yellow box of one minute. If a yellow box is already present then it is removed. If a yellow box is positioned in the time slot already allocated to an aircraft label the behaviour remains the same. However, at next step of AMAN calculation, this aircraft will be moved to the next available time slot (requiring a clearance to be sent to the pilot to speed up or slow down the aircraft to meet the new landing time slot).

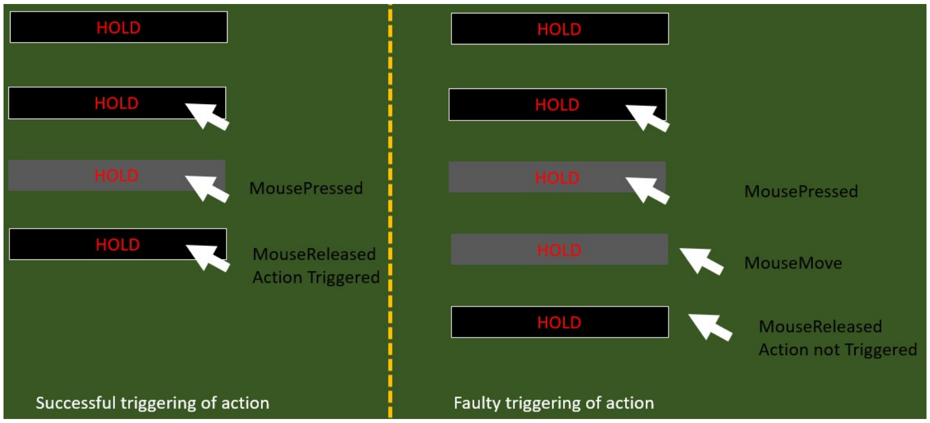


Fig. 8. Appearances of the HOLD button

3.3 Refined ATCo Tasks

With the interactive objects and interactions presented in previous sections, the task model of PLAN ATCo presented in Figure 5 has to be refined. Two sub tasks in the Manage Landing Sequence sub-goal have to be added: Zooming and moving aircraft labels. For readability of the models we present both each of the sub tasks associated to these actions and the overall model integrating them. The sub-tasks are presented in Figure 9 in which the three sub-tasks have been added between task ”Monitor LS” and task ”Change LS” which were already presented in Figure 5. The blue symbol next to the last four tasks means that these tasks are optional (i.e. it is not mandatory to perform them to reach the goal). The interleaving operator ||| means that the tasks may be performed in any order possibly starting several (or all) of them concurrently.

The overall task model is presented in Figure 10 it encompasses the preliminary task model of Figure 5 and the interaction task models.

Here we list the actions that the ATCO can execute:

- Changing the zoom level
- Changing LS
- Blocking a time period
- Putting a plane on hold

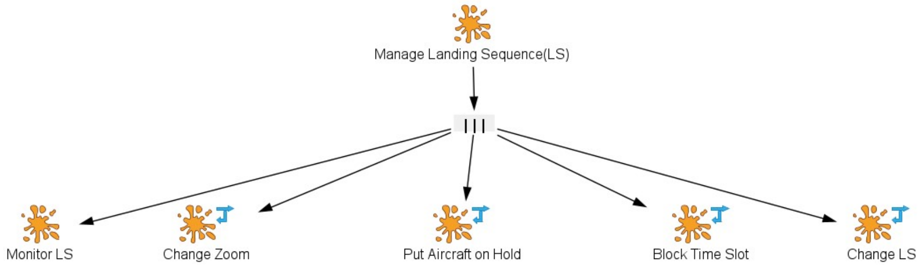


Fig. 9. Interaction tasks

4 Requirements

4.1 External events

External events affect the landing sequence produced by AMAN:

Req1 Planes can added to the flight sequence e.g. planes arriving in a close range of the airport
Req2 Planes can be removed from the flight sequence e.g. planes changing their landing airport for some reason

Interaction events also affect the landing sequence produced by AMAN:

Req3 Planes moved earlier or later on the timeline by the PLAN ATCo thus requiring from AMAN the processing of a new prediction;
Req4 Planes put on hold by the PLAN ATCo. Planes removed from HOLD will appear as normal aircrafts handled by AMAN.

4.2 Safety requirements

These safety requirements must be considered:

Req5 Aircraft labels should not overlap;
Req6 An aircraft label cannot be moved into a blocked time period;
Req7 Moving an aircraft label might not be accepted by AMAN if it would require a speed up of the aircraft beyond the capacity of the aircraft;
Req8 If AMAN is not functioning (e.g. no update after 10 seconds) the ATCo must be informed about the failure and landing sequence preparation will be done manually (without AMAN).

4.3 Automation requirements

We use here the Displays for Automated Systems requirements from the EASA Certification Specification 25 for large aeroplanes [2] with a focus on cockpits. We propose here to embed these requirements in the case study. We have mainly

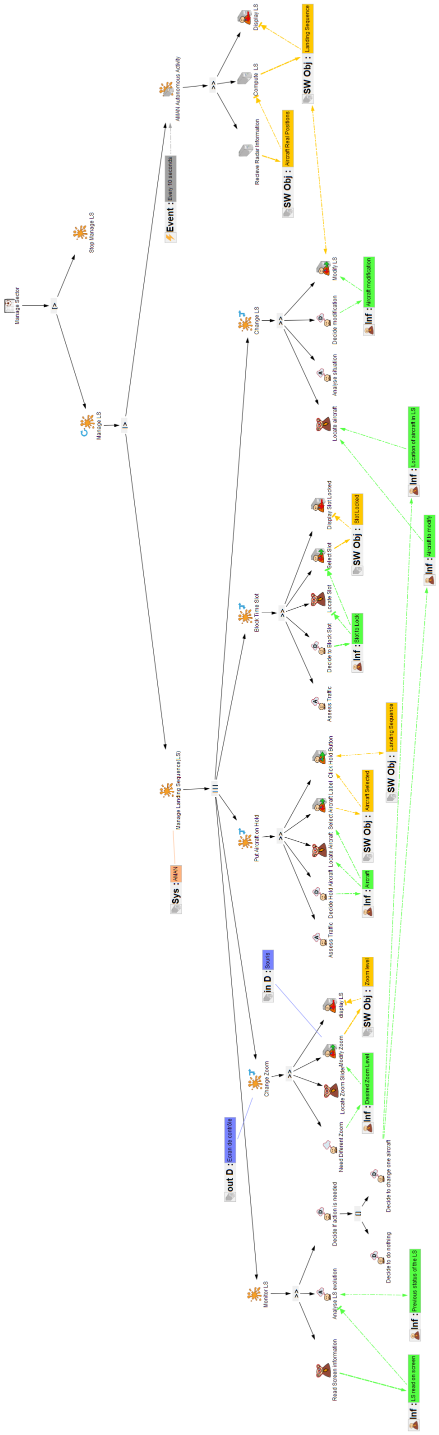


Fig. 10. Complete PLAN ATCo task model for the case study (zoom in for details)

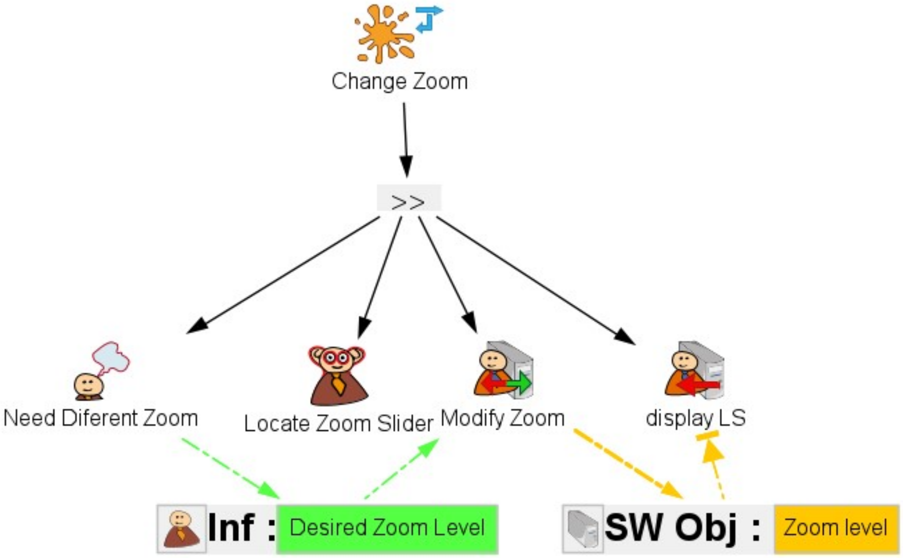


Fig. 11. PLAN ATCo task model corresponding to the changing of the zoom value

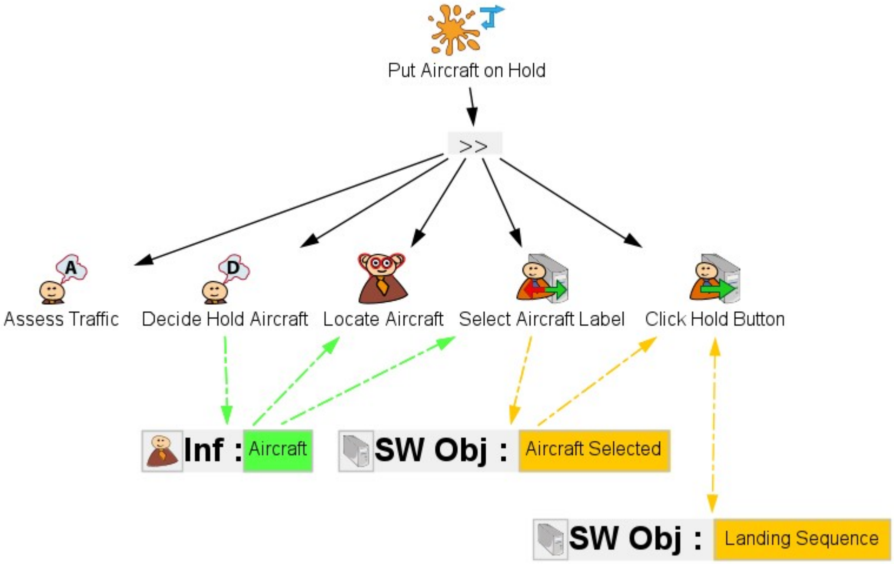


Fig. 12. PLAN ATCo task model for putting aircraft on HOLD

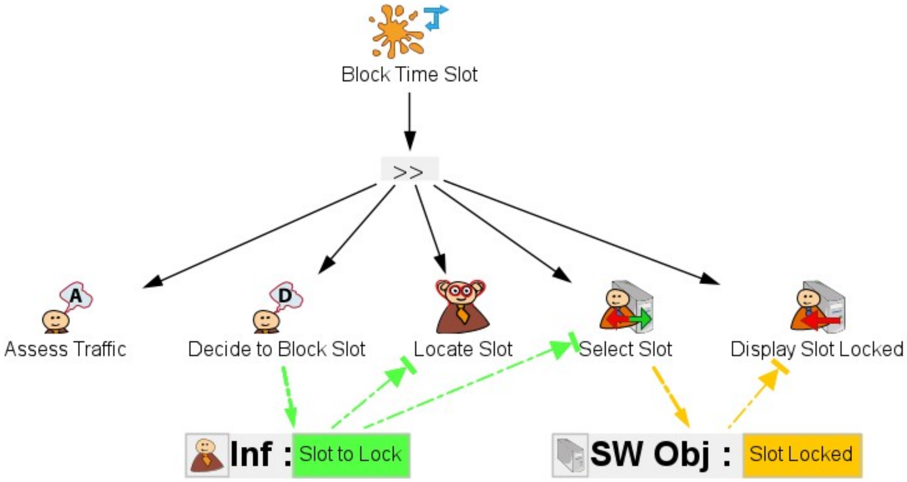


Fig. 13. PLAN ATCo task model for blocking a time slot

kept them as they are in the CS 25 and only tuned them a bit. Checking them on a given specification would be required to have a certification granted. Automated systems can perform various tasks with minimal ATCos interventions, but under the supervision of the ATCos. To ensure effective supervision and maintain ATCos awareness of system state and system "intention" (future states), displays should provide recognisable feedback on:

Req9 Entries made by the ATCo into the system so that the ATCo can detect and correct errors.
Req10 Present state of the automated system or mode of operation. (What is it doing?)
Req11 Actions taken by the system to achieve or maintain a desired state. (What is it trying to do?)
Req12 Future states scheduled by the automation. (What is it going to do next?)
Req13 Transitions between system states.

These automation requirements may be implemented in different ways on the user interface. For instance, a new scheduling of landing sequence could be presented using an animation so that the PLAN ATCo can see which changes have been made by AMAN from the previous landing sequence.

4.4 Interaction requirements

Some interaction requirements to consider are:

Req14 the set of tasks identified must be feasible on the interactive systems; this may be ensured by checking behavioural equivalence of the task model with respect to a model of the interactive application (as for instance in [8];

Req15 the HOLD button must be available only when one aircraft label is selected;

Req16 the zoom value cannot be bigger than 45 and smaller than 15;

Req17 aircraft labels must always be positioned in front of a small bar of the timeline;

Req18 Lift of the zoom slider should always be located on the slider bar

Req19 the value displayed next to the zoom slider must belong to the list of seven acceptable values for the zoom

Req20 each movement of the mouse on the ATCo table must be reflected by a movement of the cursor on the screen

Req21 there must be one and only one mouse cursor on the screen

Req22 Hold(aircraft) function can only be triggered after a mouse press and a mouse released have been performed on the HOLD button.

Req23 Hold(aircraft) function must not be triggered if there is not a mouse press and a mouse released performed on the HOLD button.

References

1. Benhacène, R., Hasquenoph, B., Cloarec, D., Favennec, B.: La gestion des arrivées en région parisienne: Aperçu général et utilisation opérationnelle de maestro. Tech. Rep. CENA/NT05-522, CENA & Eurocontrol Experimental Centre (2004)
2. EASA: Certification specifications and acceptable means of compliance for large aeroplanes (cs-25). Tech. rep., European Aviation Safety Agency (November 2021), https://www.easa.europa.eu/en/downloads/136622/en
3. Fahssi, R., Martinie, C., Palanque, P.: Enhanced task modelling for systematic identification and explicit representation of human errors. In: Human-Computer Interaction – INTERACT 2015. pp. 192–212. Springer International Publishing, Cham (2015)
4. Forbrig, P., Martinie, C., Palanque, P., Winckler, M., Fahssi, R.: Rapid task-models development using sub-models, sub-routines and generic components. In: Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds.) Human-Centered Software Engineering. pp. 144–163. Springer Berlin Heidelberg, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44811-3_9
5. Hasevoets, N., Conroy, P.: Arrival manager: Implementation guidelines and lessons learned. Tech. rep., Eurocontrol (December 2010), https://skybrary.aero/sites/default/files/bookshelf/2416.pdf
6. Kapp, V., Hripane, M.: Improving tma sequencing process: Innovative integration of aman constraints in controllers environment. In: 2008 IEEE/AIAA 27th Digital Avionics Systems Conference. pp. 3.D.1–1–3.D.1–9 (2008). https://doi.org/10.1109/DASC.2008.4702810
7. Martinie, C., Barboni, E., Navarre, D., Palanque, P., Fahssi, R., Poupart, E., Cubero-Castan, E.: Multi-models-based engineering of collaborative systems: Application to collision avoidance operations for spacecraft. In: Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems. p. 85–94. EICS '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2607023.2607031, https://doi.org/10.1145/2607023.2607031

8. Martinie, C., Navarre, D., Palanque, P., Fayollas, C.: A generic tool-supported framework for coupling task models and interactive applications. In: Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems. p. 244–253. EICS '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2774225.2774845, https://doi.org/10.1145/2774225.2774845
9. Martinie, C., Palanque, P., Winckler, M.: Structuring and composition mechanisms to address scalability issues in task models. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) Human-Computer Interaction – INTERACT 2011. pp. 589–609. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23765-2_40