18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.
Please input the preferred copyright option as mentioned in the attached "*SpaceOps-2025 Copyright Policy for Manuscripts and Presentations*"

SpaceOps-2025, ID # 21

Tracking Monitoring Activities of Operators to Ensure Operations Safety: Application to Monitoring & Control of Launcher Tracking Activities at Europe's Spaceport

Daniel Rodriguez-Hernando[a], Théo Saubanère[a], Philippe Palanque[b]*, Célia Martinie[b], Sandra Steere[a]

[a] *CSG, CNES, France, firstname.lastname@cnes.fr*
[b] *ICS-IRIT, Université de Toulouse, France, firstname.lastname@irit.fr*
* Corresponding Author

Abstract

Launcher tracking activities of operators are mainly based on monitoring information from the various components of the overall system including, the launcher itself, the antennas, the operational room, … Interactions performed by the operator on the system mainly involve neutralizing the launcher, in case of safety threats, and modifying the zoom level of some trajectory visualizations. The very nature of monitoring, which involves examining specific data from specific systems at specific times, makes it difficult to ascertain whether or not the operator has actually performed the monitoring, as these activities only take place in the operators' head. This paper presents the results of a CNES-funded research project which targets at providing assurances that the operator monitors the right information at the right time throughout the operations of launcher tracking. The proposed solution exploits interactive technologies involving an eye-tracker to capture operators gaze while monitoring. Beyond the technology, we use a task modelling notation to describe operators' tasks (both with and without such technologies) and demonstrate that tasks which exploit such technologies are able to inform the operational system about the progress in monitoring tasks as they become interactive. We show that this information can be used to train operators on monitoring activities, to provide guidance during operations and eventually to audit what happened when the operations are over. This technology has been integrated in new user interfaces for launcher tracking, like the ones presented in [35].
Keywords: Guiana Space Centre, Operations Centre (CDO), Launcher Tracking, Command and Control Operations, Monitoring, Eye tracking.

Acronyms/Abbreviations

CDO	Centre Des Operations (Operations Centre)
FSO	Flight Safety Officer

1. Introduction

The French Guiana Space Centre, Europe's Spaceport, is undergoing a massive revolution with the arrival of the new Operations Centre, the "CDO" [22] that embeds new command-and-control systems, new operational concepts and a revised organization. Beyond those changes, some of the key activities of operators envisioned in the CDO will continue to involve monitoring & control for launcher tracking activities which are critical in terms of mission and safety of populations, properties and environment on the path of the launcher.

In the context of interactive critical systems, where the stakes are high and the margin for error is minimal, the ability to track monitoring activities of operators effectively is paramount. As operators are tasked with monitoring complex and dynamic environments, the ability of the ground system to effectively track and analyse their activities is essential for ensuring operational safety and efficiency. Launcher tracking activities of operators are mainly based on monitoring information from the various components of the overall system including, the launcher itself, the antennas, the operational room, … Interactions performed by the operator on the system mainly involve neutralizing the launcher, in case of safety threats, and modifying the zoom level of some trajectory visualizations. The very nature of monitoring, which involves examining specific data from specific systems at specific times, makes it difficult to ascertain whether the operator has actually performed the monitoring or not as the activities only take place in the operators' heads.

This paper presents the results of a CNES-funded research project which targets at providing assurances that the operator monitors the right information at the right time throughout the operations of launcher tracking. The proposed approach investigates the use of eye tracking technology, alongside other input technologies, to capture and analyse monitoring activities. By focusing on the visual attention of operators, we aim to address key challenges: How can we ensure that operators are looking at the right information at the right time? How can we track their progress in executing tasks and ensure that they do not overlook critical information? To tackle these questions, we have explored various interactive technologies, input devices and interaction techniques, that can enhance the monitoring process exploring their impact on the addition of cognitive load or distractions. Beyond the selection of input devices (such as eye-tracker, mouse, keyboard, …), the selection and the design of appropriate interaction methods is crucial as they must facilitate the operator's work while ensuring that the monitoring system remains easy to use and efficient.

To illustrate our findings, this paper presents a case study involving the interactive application of command-and-control of the FSO (Flight Safety Officer). We use a task modelling notation to describe FSO operators' tasks with and without such technologies. We use these task models to demonstrate that tasks which exploit this technology make it possible to inform the operational system about the progress in monitoring tasks. We show that this information can be used to train operators on monitoring activities, to provide guidance during operations and eventually to audit what happened when the operations are over. This technology has been integrated in the new user interfaces for launcher tracking like the ones presented in [35].

The paper is structured as follows. Section 2 presents more precisely the context of Flight Safety operations and describes the envisioned system for the CDO at the French Guiana Space Centre, Europe's Spaceport. Section 3 presents the task modelling notation HAMSTERS used to describe operators' tasks and its use for the description of FSO tasks with a special focus on the monitoring tasks. This section also highlights the importance of tracking how the monitoring activities are performed as they may impact safety. Section 4 provides an analysis of existing research work and technologies for tracking users' monitoring activities. While there is a classical focus on hardware input devices (e.g. eye-trackers) it highlights the importance of the interaction technique associated to the device. This section also explains the importance of the use of multimodal technologies and how they can improve overall usability. Additionally, this section presents the solution that has been selected for tracking FSO monitoring activities. It presents the input devices and their associated multimodal interaction technique which has been designed to ensure both usability of the interactions and the safety of operations. Section 5 discusses the benefits and the limitations of the proposed contribution, leaving some elements to future work. Section 6 concludes the paper and presents directions for future work.

2. Context: the Flight Safety Command and Control System

2.1 The CSG future spaceport and the CDO

The new operations centre is profoundly renewing all Core Launch Range systems and services, including all the IT and operational equipment that is necessary to support the execution of the launch campaigns. The key objectives are to boost launch frequency by decreasing the time required to reconfigure the entire spaceport, all while ensuring operational safety. To accomplish this, it is essential to optimize the organizational structure and the operators work, as well as enhance training programs, in order to launch different vehicles in a short period. Along with this renewal, Flight Safety Operations are changing to adapt them to the arrival of the new "classic" launch vehicles like Ariane 6 [1] or Vega-C [23], but also reusable vehicles like CALLISTO [2] and other micro vehicles.

2.2 The Flight Safety Officer missions

Flight Safety Operations belong to the group of critical operations of the spaceport and can be decomposed into several missions carried out to ensure the safety of populations and properties, the protection of public health, and the protection of the environment on the Earth's surface against any damage that may result from the in-flight operation of the launch vehicle. These Flight Safety Operations are subject to the respect of the French law that governs space activities in France, i.e. "the French Space Operation Act" [5], and the derived document "Order Regulating the Operation of Installations of the French Guiana Space Centre" [6]. According to these laws, several missions, like the "Safety and intervention mission" (also called MSI, described in article 63 of [6]) or the "Safety and alert mission" (also called MSA, described in article 64 of [6]), must be carried by the Flight Safety Operations team. This team is composed of several operators, each having a particular role, and associated tasks. In this paper we focus on the tasks that the Flight Safety Officer (FSO) must perform, throughout the MSI, with their command-and-control monitoring system, beginning with the lift-off of the launch vehicle and continuing until the MSI's conclusion.

The launch mission surveillance typically lasts around 500 seconds (i.e. about 8 minutes). In that period of time, at any time the FSO:

- Is required to perform multiple tasks concurrently (more detailed in subsection 2.2), a substantial portion of the operator's work involving real-time monitoring tasks;
- Must assess the potentially dangerous nature of the launch vehicle in flight:
  - by monitoring its horizontal and vertical trajectory;
  - by monitoring possible deviations from the pre-calculated trajectory;
  - by monitoring the status of the resources contributing to this mission;

- Must intervene if necessary to neutralize the launch vehicle
- Must deactivate the onboard neutralization system when the total missions time (defined at mission preparation) has elapsed.

2.3 The FSO User Interface Prototype

This section presents a user interface prototype of an FSO interactive command-and-control system. In this user interface prototype, elements of various sizes and shapes, including composite elements (elements nested within other elements) coexist, which the FSO must monitor during the MSI. For the purposes of this paper, and due to confidentiality concerns, the user interface prototype presented in this section is a simplified version of the actual user interface used by the operator. However, it is sufficiently representative of the layout and visual appearance of the original design.

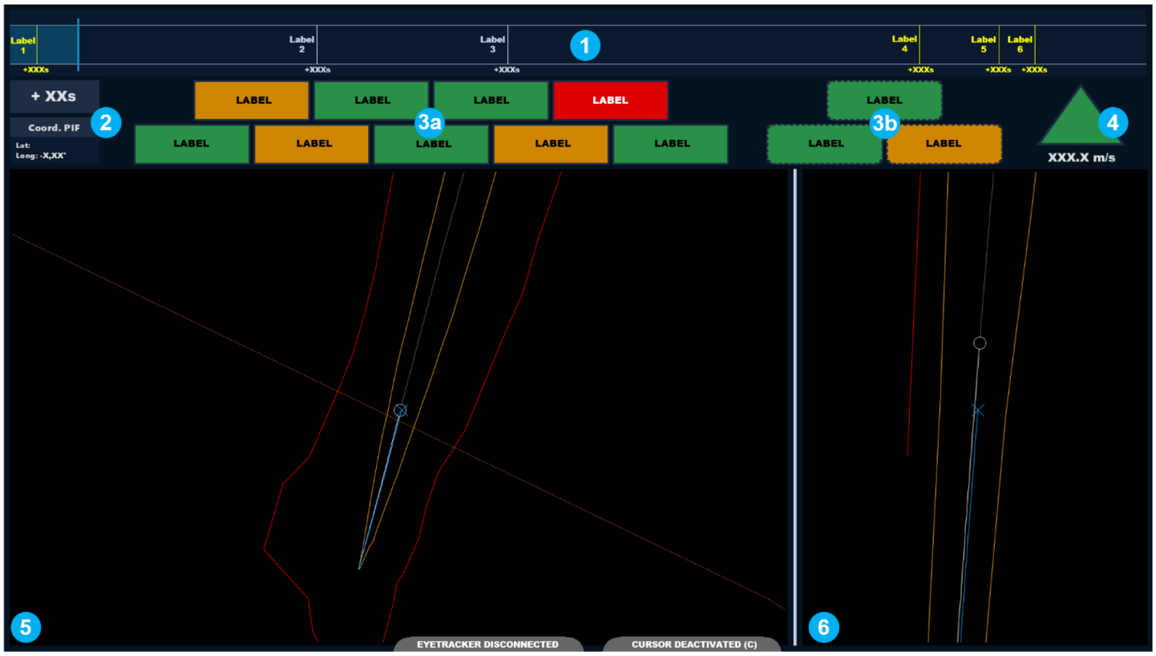

Figure 1. Prototype of FSO user interface

This user interface allows the FSO to perform monitoring activities and is composed of several parts:

- The mission timeline displaying the planned events during the launch, like for example the separation of a launch vehicle stage (Figure 1, disk 1). This timeline supports the operator task "Monitor the mission timeline" in task model in Figure 2.
- The current time and the launch vehicle detailed coordinates (Figure 1, disk 2), which support the operator task "Monitor mission-related information".
- The status indicators for the systems and parameters that contribute to the overall functionality of the command-and-control system (Figure 1, disks 3a and 3b). These status indicators are organized geographically within the interface in two groups, based on their importance. The left group are the most important ones, as they present all the critical systems and parameters on which a failure could imply the neutralization of the launcher. On the other side, the right group (Figure 1, disk 3b) presents systems and parameters the FSO uses to anticipate launch vehicle dangerousness. Both groups of status indicators support the operator task "Monitor contributing resources" in task model in Figure 2. A failure in any of the systems presented by these indicators is displayed on the UI by changing the colors of the tiles (green means that the corresponding system is nominal, orange means that the system presents a non-mission-critical failure while red means that the system in not functional).
- The vertical speed triangle (Figure 1, disk 4) which displays the vertical speed of the launch vehicle, and support the task "Assess the vertical speed of the launch vehicle" in task model in Figure 2.
- The launch vehicle's horizontal position area (Figure 1, disk 5), including the latitude (y-axis of the left graph) and longitude (x-axis of the left graph) of the launch vehicle's horizontal position. This area

supports the operator task "Monitor the horizontal trajectory" in task model in Figure 2 and in task model in Figure 3.

- The launch vehicle's vertical position area (Figure 1, disk 6), including the altitude (y-axis of the right graph) and the curvilinear distance from the launch pad (x-axis of the right graph) of the launch vehicle's vertical position. This area supports the operator task "Monitor the vertical trajectory" in task model in Figure 2.

3. Description of Operator's Tasks Including Monitoring

This section presents the FSO operator's tasks and highlights the specific issues when it comes to analyse the work of this operator. We first introduce the HAMSTERS notation which has been designed to describe users' work and then apply it to describe the work of the FSO.

3.1 The FSO Tasks

A detailed description of FSO's tasks is needed to understand to understand precisely the type and quantity of work FSO has to perform during the mission. Given the extensive nature of the operators' tasks, the task models are inherently large. Figure 2 provides an excerpt from the task model associated with the goal of "Ensuring Flight Safety During the MSI," specifically focusing on the abstract task "React to the launch vehicle dangerousness status".

This task model was edited using the HAMSTERS notation and tool [17]. Using that notation, a task model is a tree which nodes are either a task (Table 1) or a temporal ordering operator (Table 2). A given task may be an abstract task (to represent the fact that the user may have to reach several intermediate sub-goals to be able to reach the main goal), a user task, or an interactive task (depicted in Table 1). Refined types of user tasks include motor, perceptive, and cognitive (depicted in the "User" row in Table 1). A cognitive task can also be refined into a cognitive analysis task, a cognitive decision task, or a cognitive prediction task. Interactive tasks are used to represent tasks which are performed on an interactive system such as the FSO command-and-control system. These task types make it possible to describe precisely the FSO work and to analyze it, for instance, by computing the cognitive complexity of the work [26].

Table 1. Task types in HAMSTERS notation

Category of task type	Task types	
	Abstract	Concrete
Main goal	Main goal	Not applicable
Abstract	abstract task	Not applicable
User	abstract user task	perceptive task; motor task; cognitive task; cognitive analysis task; cognitive decision task; cognitive prediction task
Interactive	abstract interactive task	Interactive input task; Interactive output task; Interactive input/output task

Table 2. Types of operators in HAMSTERS task models

Operator type	Symbol	Operator type	Symbol
Enable (sequence)	>>	Disable (deactivation)	[>
Concurrent	|||	Suspend-resume	|>
Choice	[]	Order Independent	|=|

HAMSTERS also enables the description of data (e.g., information, knowledge, and objects), devices (e.g., input device, output device), and systems required or used in order to perform a task. Information, knowledge, and objects (which can be physical objects or software objects) are depicted using labels preceded by the abbreviation of the type of data. Information related to knowledge required to perform a task is very useful to assess the training needs and to build the training material (in the context of satellite ground segments [27] and in civil aviation [28]).

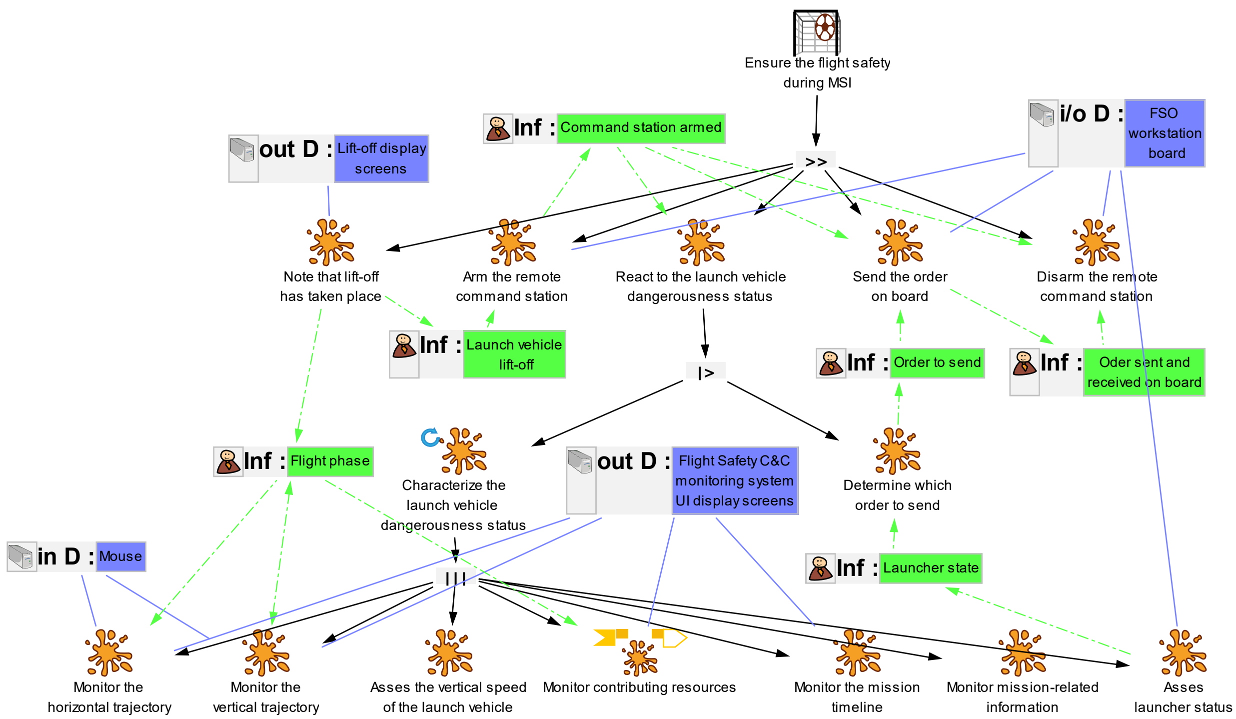

Figure 2. Abstract view of the task "Ensuring Flight Safety During the MSI"

The task model in Figure 2 presents an abstract exhaustive description of FSO's work during the MSI. The model must be read from top to bottom and from left to right. The main goal is to "Ensure the flight safety during the MSI". Taking as a reference point the abstract task "React to the launch vehicle status", this abstract task can be further detailed through the abstract task "Characterize the launch vehicle dangerousness status" which may be disabled by the abstract task "Determine which order to send". The abstract task "Characterize the launch vehicle dangerousness status" refines in the following concurrent abstract tasks: "Monitor the horizontal trajectory", "Monitor the vertical trajectory", "Assess the vertical speed of the launch vehicle", "Monitor contributing resources", Monitor the mission timeline", "Monitor mission-related information" and "Asses launcher status".

Figure 3 shows an excerpt of the refinement of the abstract task "Monitor Contributing Resources". To monitor the systems and parameters that contribute to the overall functionality of the command-and-control system, the FSO must assess various status indicators. As depicted in Section 2.3, these indicators are categorized into two groups based on the significance of the information they convey. The left group comprises the most critical status indicators, which provide a synthesis of the state of critical systems and parameters that the operator must consider to form an accurate mental model of the launch vehicle dangerousness state (based on a set of criteria established by the Flight Safety team), and to neutralize the launcher if necessary. Conversely, the right group functions as alerts, enabling the operator to anticipate potential risks associated with the launch vehicle. A failure in any of the systems concerned by these status indicators is presented on the UI by changing the colors of the indicators (green means that the corresponding system is nominal, orange means that the system presents a non-mission-critical failure while red means that the system in not functional). The excerpted task model presented in Figure 3 details the operator's monitoring of the top-left status indicator, from the left group, where it can be appreciated that the majority of the operator's tasks are perceptual and cognitive in nature, indicating that the system lacks the capability to assess the operator's progress in completing these tasks.

In addition to monitoring contributing resources, the FSO performs several activities (not presented here) in parallel to characterize the dangerousness state of the launch vehicle (abstract tasks presented at the bottom in Figure 2). For example, the abstract task "Monitor horizontal trajectory" is meant to monitor the current horizontal position and trajectory of the launch vehicle, and to compare it mainly with a geographical limit beyond which the launcher must be neutralized. All of these tasks require the analysis, decision-making, and projection of many pieces of information that the FSO perceives from the user interface. These tasks represent a big portion of all the tasks the FSO must do during the MSI, implying a significant cognitive workload, particularly given the rapid response required within the limited timeframe of 500 seconds.

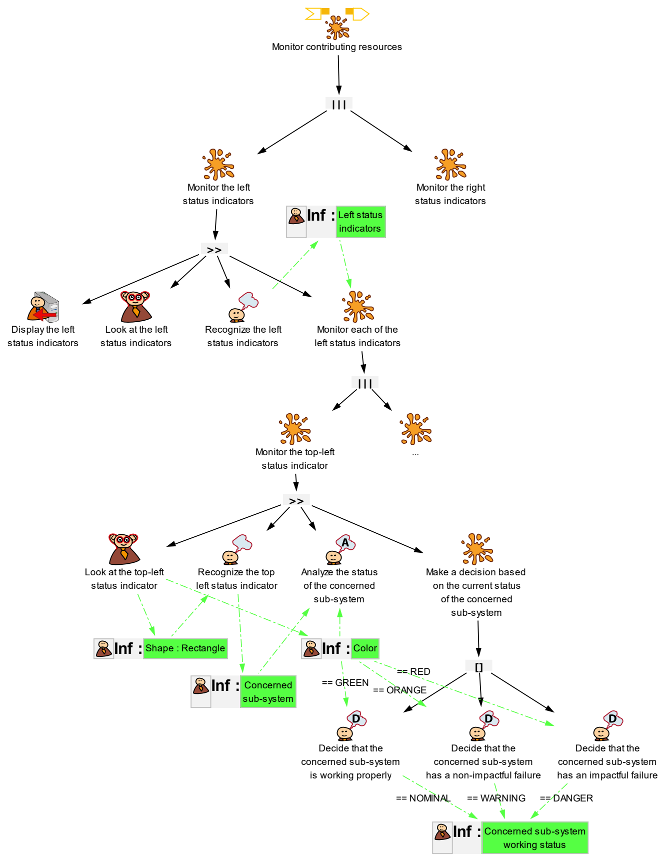

Figure 3. Excerpt task model from the task model "Monitor Contributing Resources"

3.2 The Problem of Tracking Operators' Performance of Monitoring Tasks

The task model in Figure 3 only uses a subset of the task types proposed by HAMSTERS; the perceptive and the cognitive tasks. This is understandable as, as explained above, monitoring tasks mainly consist in acquiring information from a system or the environment and analysing it before making a decision. It is thus difficult to check that the monitoring is actually performed and that it is performed as required.

As of today, an operator designated as the FSO must complete a specialized training program and achieve qualification prior to being assigned to a launch vehicle campaign. The training program is structured around the tasks operators are expected to perform during the MSI. During a training session, the trainees execute their tasks and verbally communicate their assessments of the launch vehicle's dangerousness state. The trainer observes and listen to the trainees. The observations primarily focus on the verbal exchanges, allowing trainers to evaluate whether trainees have learned the necessary flight safety criteria for accurately assessing the dangerousness state of the launch vehicle, and that they can apply them effectively. However, while verbal communication provides valuable insights about the trainee's capabilities, it often lacks critical information regarding whether the trainee is monitoring the appropriate information at the right moment during the MSI. While this issue is key at training time, it is also an issue during operations and even after operations if the activity of the operators has to be checked.

With this respect, the work presented in this paper proposes a solution to this problem by 1) adding interactive technologies to the command-and-control system in order to make interactive monitoring tasks feasible 2) transforming monitoring work into an interactive monitoring one while reducing to the minimum the evolution of the operators' tasks.

4. A Practical Multimodal Solution to Track FSO Monitoring Activities Using Interactions

In this section, we present the solution that has been designed for ensuring the tracking of FSO monitoring activities. This solution combines an eye-tracker and mouse as input devices and required the design of an evolution of the Gaze-Lock interaction technique in order to cope with the specificities of the FSO work (which embeds specific operational constraints including strong real-time interactions). The solution targets effective and efficient communication between the FSO and the underlying command-and-control interactive system.

4.1 Overview and Selection of Input Devices for Tracking FSO Monitoring Activities

To effectively enable interactive monitoring, input devices and their associated interaction techniques must support both pointing and activation tasks. This means that the input device should be able to make explicit where the operator is looking at and to allow the operator to explicitly confirm that information, gathered by the input device. While the literature has extensively examined specialized pointing and activation input devices and interaction techniques, ranging from traditional mice and trackpads [12] to advanced technologies such as eye tracking systems and gesture recognition [20], their applications differ significantly across various domains, including accessibility [4], gaming [24], and critical environments like civil aircraft cockpits [7] or military aircraft one [9]. In the space domain the use of multimodal interaction has been reported for satellite ground segment but the operational constraints are far from real time [8]. Despite this diversity, we found no studies proposing input devices or interaction techniques specifically designed for interactive monitoring, allowing pointing and activation of User Interface (UI) elements in real-time monitoring in any context of use.

To address this gap, we have explored input devices taking into account, as much as possible, the following constraints:

- Their exploitation can be performed on the existing FSO user interface which has already been validated with operators
- They must be cost-effective and minimally disruptive to the operators' workload and environment.
- They device should favor comfort and usability to reduce strain and fatigue during the MSI's 500-second monitoring interval.
- They must correspond to the context of work of the operators i.e. static position in front of a static monitor.
- They must offer easy to use APIs (Application Programming Interfaces) to make their integration in terms of programming as simple as possible.

Consequently, we selected two input devices, the eye-tracker and the mouse:

- **Eye-Tracker**: This device exploits cameras and infrared light to monitor the position and movement of the eyes. Integrating eye tracking into monitoring systems offers several advantages, including enhanced user-in-the-loop engagement through natural interaction [14], allowing users to point to items simply by looking at them, thereby streamlining the interaction process. Furthermore, eye tracking provides valuable insights into user attention and behavior, enabling designers to optimize interface layouts based on gaze patterns. Activation is an issue with an eye-tracker as the main interaction technique called dwell requires the operator to look in the same location for an arbitrary amount of time (usually 300ms) [14]. This was not fitting the operational constraints of the FSO and we thus decided to use a second input device for the activation task.

  Regarding the type of eye-trackers, there are two primary types: screen-based and wearable. Screen-based eye-trackers are typically integrated into monitors, making them suitable for stationary environments, while wearable eye-trackers offer greater flexibility for dynamic settings. Given the stationary nature of the FSO's work environment and the need to minimize disruption, we preferred screen-based eye-trackers.

  From a technological perspective, several recent commercial screen-based eye-trackers could fulfill our requirements; however, the eye tracking device chosen for its adherence to all previously outlined constraints is the Smart Eye AI-X [21] from Smart Eye and iMotions. This device was chosen primarily for its multiple control and programming interfaces, including a remote-control interface, an event reception interface, and an event transmission interface. These interfaces facilitate the transmission of various levels of information regarding gaze movement capture, ranging from raw data (directly receiving the coordinates of each eye on the screen) to more abstract levels (such as invoking a procedure to add a new user).
- **Mouse**: The mouse is a widely deployed input device that detects movement relative to a flat surface, requiring users to move the mouse to point to a location on the screen. We selected the mouse due to its standardized use in activation tasks, particularly since it is already employed by operators during the MSI

and is available on the interactive command-and-control system. However, pointing tasks with the mouse would induce fatigue and require a significant movement on the table, potentially interfering with the operations.

- **Multimodality**: as each input device (the eye-tracker and the mouse) present complementary drawback and benefits, we have decided to exploit them in a multimodal way for tracking monitoring activities:
  - the non-intrusive eye-tracker for pointing (the gaze of the operators corresponds to the objects they are monitoring).
  - the already available and known mouse for activation i.e. confirmation that the object detected by the Eye Tracker has effectively been monitored.

4.2 The importance of interaction techniques

Interaction techniques, as defined in [13], are the integration of input and output encompassing all hardware and software elements that enable users to accomplish low-level tasks. These techniques are fundamental to the efficiency of user interfaces and must be carefully designed to align with those specific tasks, their contexts of use, and the users' needs. For example, an interaction technique that is effective for data entry (typically WIMP interaction techniques [29]) may not be suitable for real-time monitoring tasks, where speed and accuracy are critical (where multimodal interactions with real-time constraints are needed [9]).

The tasks of interest to us—pointing and activation—can be split into three fundamental interaction phases: acquisition, selection, and activation [18]. Each phase involves distinct cognitive and motor processes and may benefit from different interaction design strategies. The acquisition phase entails directing the pointer (the mouse cursor) or gaze toward the target, selection confirms that the current target is the correct one, and activation initiates an action on that target. Poorly designed techniques often blur these phases, leading to mis-selections or premature activations. A clear distinction and appropriate control at each stage are especially critical when designing for emerging input modalities. This is particularly relevant for eye tracking based interaction techniques, where challenges such as the "Midas Touch" effect [16] (unintended activations due to the natural tendency of the eye to scan) can significantly hinder usability. Gaze-hold mechanisms (frequently Eye-Dwell) which require maintaining gaze on a target for a specified duration before selection and even activation, aim to mitigate this issue but may introduce latency and fatigue [19]. Achieving the right balance between responsiveness and control is a complex challenge, underscoring the necessity of fine-tuning interaction techniques. Minor adjustments in timing, sensitivity, or feedforward and feedback mechanisms can dramatically influence the effectiveness of an interaction method [30]. This is particularly true in the context of pointing and activation, where precision and timing are crucial, and user expectations are shaped by millisecond-level interactions. For this reason, we selected to adapt and extend the Gaze-Lock interaction technique [25] (to support multimodal interaction with the mouse), which is presented in the next section. As demonstrated in [31] in order to cope with the complexity of interaction techniques, their description requires formal and unambiguous modeling techniques.

4.3 The Extended Gaze-Lock Interaction Technique

Shaoyao Zhang et al. [25] introduced the Gaze-Lock interaction technique, which is a multimodal eye-hand interaction method that utilizes an eye-tracker for the acquisition and selection phases, while employing the Logitech MX ERGO [15] trackball for activation. In the present study, we propose the use of a mouse as the activation input device, leveraging the device already employed by the FSO during the MSI for zooming in and out on the horizontal and vertical position areas (Figure 4, disks 5 and 6). In the remainder of this paper, we will refer to an interactive element of the FSO user interface as AoI (Area of Interest).

Figure 4 presents the behavior of the Gaze-Lock interaction technique with an AoI in the FSO user interface. Figure 5 presents examples of visual rendering of an AoI depending of the current state of the interaction technique. The selection of an AoI requires the FSO's gaze to first enter the designated AoI (illustrated in Figure 4 and Figure 5, disks 2). If the operator's gaze is detected outside the AoI for a minimum of 7 instances (not necessarily consecutive, as shown in Figure 4 and Figure 5, disks 2), the AoI reverts to its initial state. Conversely, if the operator's gaze is detected within the AoI for at least 7 instances (again, not necessarily consecutive, as depicted in Figure 4 and Figure 8, disks 2), the AoI becomes locked, centering the cursor that indicates the user's gaze position on the screen within the AoI (Figure 7 and Figure 5, disks 3). In the locked state, there are two possibilities. First, if the operator's gaze is detected outside the AoI for an additional 7 instances (for a total of 14 instances, again not necessarily consecutive, as illustrated in Figure 4 and Figure 5, disks 3), the AoI is unlocked, returning to its initial state, and the cursor shifts to the current gaze position.

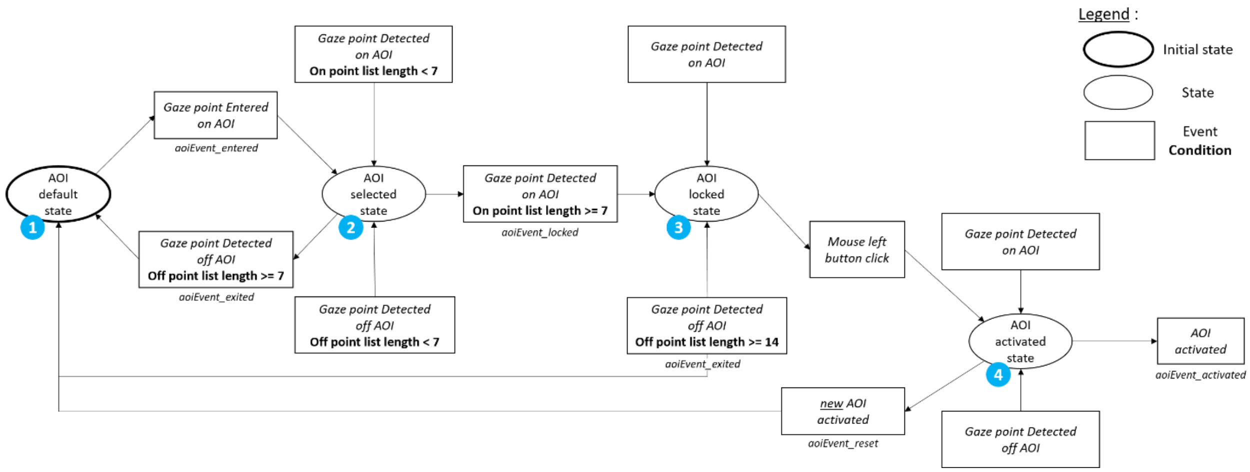

Figure 4. Simplified automaton of the Gaze-Lock interaction technique

Second, if the operator clicks the left mouse button, regardless of the mouse cursor's position (Figure 4 and Figure 5, disks 4), the AoI becomes active, meaning that the operator monitored it. Following activation, the AoI can only revert to its initial state upon the reception of an activation event (aoiEvent_activated) from another more recently activated AoI.

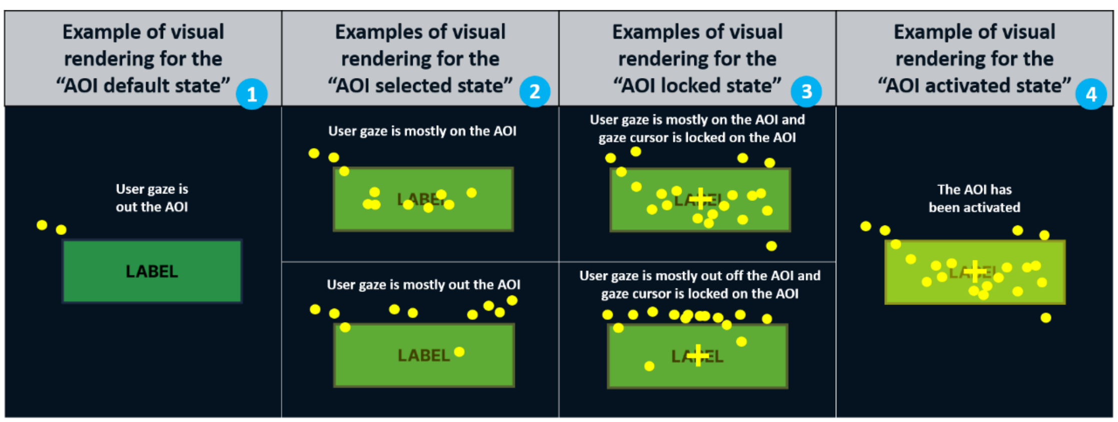

Figure 5. Examples of visual rendering for each state of the Gaze-Lock interaction technique

In order to achieve the optimal selection duration of approximately 100 ms, as suggested in [25] introduced, it is crucial to take into account the technical specifications of the input device utilized for this interaction technique, the Smart Eye AI-X introduced in Section 4. This eye-tracker operates with a refresh rate of 16 ms for capturing the user's gaze. Consequently, to facilitate the selection of an AoI within the target duration of approximately 100 ms, the operator's gaze must be detected at least 7 times during the selected and locked states to determine the user's intent to lock or unlock the AoI.

4.4 Evolution of FSO Tasks when Exploiting the Extended Gaze-Lock Interaction Technique in Monitoring Work

By integrating new interactive technologies to the command-and-control system, the tasks of the FSO are modified. This refers to the well-known task-artifact lifecycle proposed by Carol and Rosson [10], which was concretized using systems and tasks models in [11]. We demonstrate this by making explicit the changes within the task model presented above.

Figure 6 presents an excerpt from the task model "Monitor contributing resources", previously illustrated in Figure 3, incorporating two abstract tasks: "Monitor the alert filter displayed on the top-left status indicator" and "Capture the monitoring of the top-left status indicator". The first abstract task, "Monitor the alert filter displayed on the top-left status indicator", encompasses the tasks the operator must do to perceive and consider a guidance mechanism, displayed by the system following the triggering of the event "Alert Filter Displayed". This mechanism, which will be elaborated upon in the next section, is designed to get the operator's attention to critical information that has not yet been acknowledged.

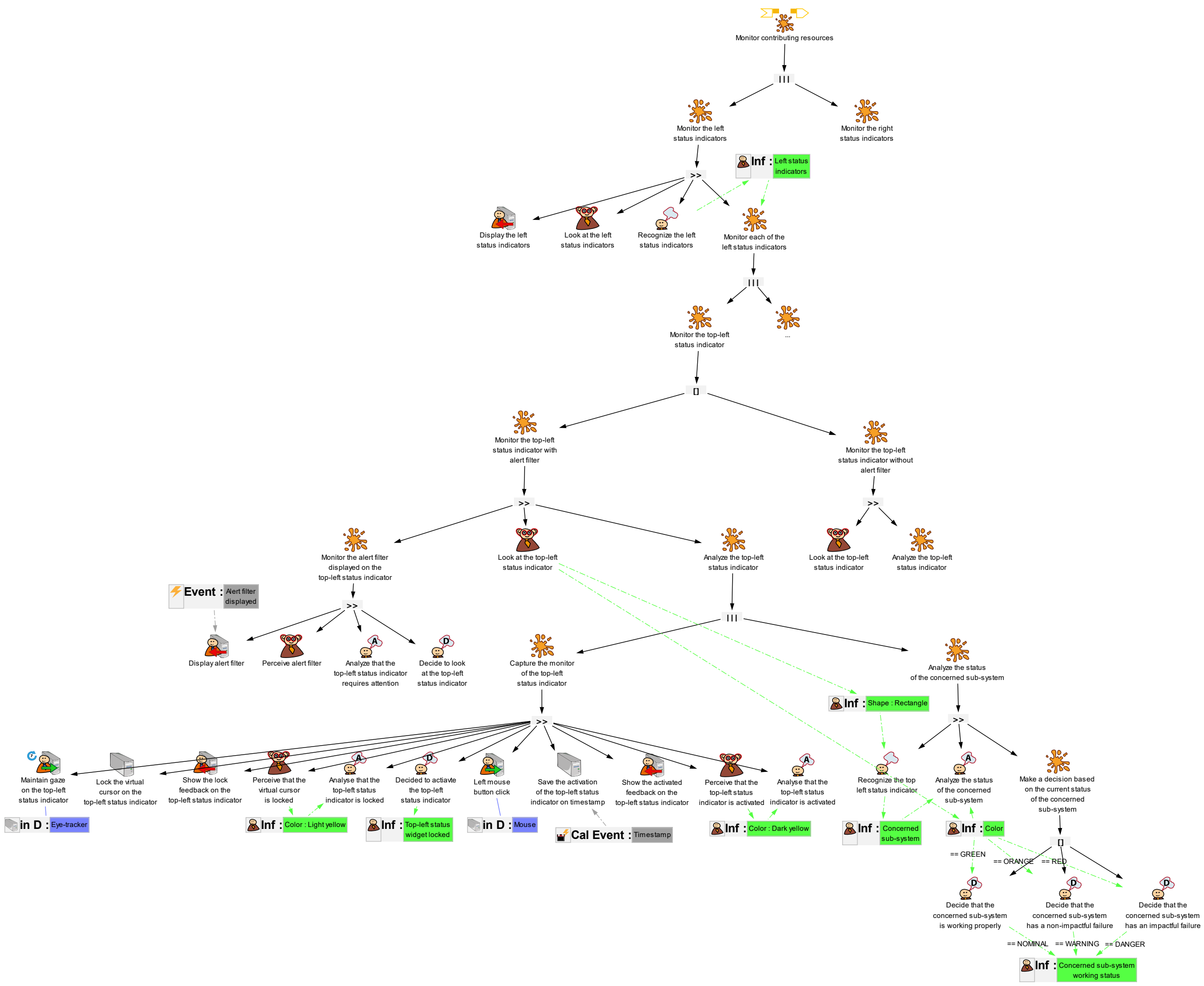

Figure 6. Excerpt task model from the task model "Monitor contributing resources" with integrated interactions

The second abstract task, "Capture the Monitoring of the Top-Left Status Indicator," encompasses all user, interactive and system tasks necessary for the operator-system duo to determine where the FSO is looking and at what moment. In this abstract task, utilizing the Gaze-Lock interaction technique described in the previous section, the operator must maintain their gaze on the top-left status indicator, designated as the AoI, until a low-opacity yellow filter rectangle appears on the top of the AoI and the virtual cursor is locked onto it (Figure 5, disk 3). Once the AoI is locked, the operator must confirm their perception of the information displayed by clicking the left mouse button. This process enables the system to accurately record the specific user interface element the operator was observing and the corresponding time, thereby assessing the perceptual aspect of their monitoring activities.

5. Benefits and Limitations of Adding Tracking Capabilities to Monitoring Activities

This section presents the benefits and the drawbacks of the solution presented in the paper.

5.1 Benefits of Adding Tracking Capabilities to Monitoring Activities

As explained in the paper, the main benefit of the proposed solution is to be able to track the work done by the operators when they perform their monitoring activities. However, the goal is not to do so but to exploit it for improving operations safety.

We have envisioned three benefits of exploiting these interactive technologies:

- **Training**: the technology can be used to track trainees' activity and learning progress while they perform a training. Indeed, the gaze trajectory can be easily monitored, as well as their understanding of the work to be done during monitoring;
- **Guidance**: the technology can be used to provide guidance at runtime to operators (for instance showing to them where to look on the user interface when they are not looking at an information that requires attention). Such user interface guidance mechanism could be designed as an attention getter, as the RedAlert technique proposed for Air Trafic Controlers [32];
- **Post-operations auditing**: The monitoring activities of the operators could be checked and audited when the operations are over. This would help in identifying the needs for the redesign of the user interfaces (e.g. if an information is frequently overlooked or response time is too high).

5.2 Limitations of Adding Tracking Capabilities to Monitoring Activities

Adding the Gaze-Lock interaction technique to an existing command-and-control system reveals disadvantages with respect to "more standard" operations. This includes the following criteria:

- Workflow disruption (effectiveness): how the interaction technique affects the workload in terms of input interactive tasks (how much effort it takes to perform an action) but also output interactive tasks (how much effort it takes to perceive and understand the interaction technique feedback).
- Performance (efficiency): the estimated selection and activation duration in comparison with the original interaction technique proposed in the literature.
- Learning curve: how easy/difficult it is to master the interaction technique and how this could affect training programs (by adding lessons exclusively for the trainee to learn how to use an interaction technique, etc.).
- Reliability: to what extent the interaction technique is reliable and consistently works as expected.

In terms of workflow disruption, the extended multimodal Gaze-Lock interaction technique introduces a workflow disruption by requiring additional interactive input tasks i.e. an explicit, voluntary left mouse button click to activate the AoI and confirm that the operator's focus has been correctly detected. Beyond this additional interactive input task, the operator has to perceive and analyze the opacity of the yellow filter rectangle on the AoIs to determine whether an AoI has been selected or activated, thereby introducing several perceptive and cognitive tasks into the operator's workflow.

In terms of performance, the technical characteristics of the input device directly influence the interaction technique, as well as the size and distribution (layout) of the AoIs within the user interface. For the eye-tracker interaction techniques, Gaze-Lock allows for this process in about 100 ms. The AoI size and distribution on the user interface have a clear impact on the performance of interaction techniques [19]. In this work, one key requirement was to be able to integrate the interaction techniques into the existing operator user interface. However, would such interaction technique be deployed, these parameters should be considered to tune both the interaction technique and the user interface elements.

Regarding the learning curve, the interaction technique will require training for usage and to learn the feedback it provides to the operator. Gaze-Lock adds a gaze cursor locked at the center of the AoI, which facilitates focus on the selected AoI but results in a visual jump of the gaze cursor when the AoI is unlocked. Additionally, Gaze-Lock may lead to situations where operators divert their attention to feedback states rather than focusing on critical information. It is crucial to note that the learning curve should be considered when developing training programs, as one or more training sessions may be necessary for effective and efficient usage.

In terms of reliability, the use of a formal notation at design time enabled to precisely specify the behavior of the Gaze-Lock interaction technique. Such a formal notation supports the verification of the properties of the interaction techniques. For example, from the automata presented in Figure 4, we can prove that in whatever state the interaction technique is in, there is always a sequence of actions to bring the interaction technique back to its initial state. However, the reliability of the programmed and running interaction techniques then requires to apply programming principles that help to ensure that the produced software matches the specified automata (e.g. traceability from the specification to the software). Another option is to use an integrated modeling and development environments that enable to formally specify the behavior of the interaction technique and execute it within a user interface [3]. We have already performed this type of model-based implementation [33] while developing touch-based interaction techniques resilient to turbulences in aviation and also demonstrated the existence of desired properties in the models [34].

6. **Conclusion**

This paper has presented the issues related to tracking how operators perform monitoring activities as no information is provided to the command-and-control system they use. We have proposed a solution exploiting an implicit input device (an eye-tracker) in combination with an explicit input device (a mouse) to inform the command-and-control system about the monitoring actions which are being performed.

This new interaction technique has been applied to the user interface of Flight Safety Officers for launch tracking operations at the CNES space center in French Guiana.

The paper has also highlighted the fact that the solution is generic and can be applied to any command-and-control system as long as it is compatible with the deployment requirements of an eye-tracker.

Future work involves designing specific user interfaces and identifying guidelines for user interfaces that ensure full compatibility with such new interaction techniques.

Acknowledgements

This work was partially funded by the project CNES 21055/00.

References

[1] Ariane 6 launch vehicle webpage, https://cnes.fr/en/projects/ariane-6, last accessed December 2024.
[2] CALLISTO launch vehicle webpage, https://cnes.fr/en/projects/callisto, last accessed December 2024.
[3] J. C. Campos, C. Fayollas, M. D. Harrison, C. Martinie, P. Masci, and P. Palanque. Supporting the Analysis of Safety Critical User Interfaces: An Exploration of Three Formal Tools. ACM Trans. Comput.-Hum. Interact. 27, 5, Article 35 (October 2020), 48 pages. DOI: https://doi.org/10.1145/3404199
[4] M. Cicek, A. Dave, W. Feng, M. X. Huang, J. K. Haines, and J. Nichols. Designing and Evaluating Head-based Pointing on Smartphones for People with Motor Impairments. In Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20), 2020, Article 14, 1–12. https://doi.org/10.1145/3373625.3416994
[5] CNES. French Space Operations Act. N° 2008-518, 2008, 3rd June.
[6] CNES. Order Regulating the Operation of Installations of the French Guiana Space Center. CNES/P/2010-1, 2010, 9th December.
[7] A. Cockburn, C. Gutwin, P. Palanque, Y. Deleris, C. Trask. Turbulent Touch: Touchscreen Input for Cockpit Flight Displays. In: International Conference for Human-Computer Interaction, CHI'17, 2017, 6742-6753.
[8] M. Ould, Bastide R., Navarre D., Palanque P., Rubio F., Schyn A. "Multimodal and 3D Graphic Man-Machine Interfaces to Improve Operations". Eighth International Conference on Space Operations, Montréal, Canada, May 17 - 21, 2004.
[9] R. Bastide, D. Navarre, P. Palanque, A. Schyn & P. Dragicevic. A Model-Based Approach for Real-Time Embedded Multimodal Systems in Military Aircrafts. Sixth International Conference on Multimodal Interfaces, ACM ICMI'04, October 14-15, 2004 Pennsylvania State University, USA.
[10] J. M. Carroll and Mary Beth Rosson. 1992. Getting around the task-artifact cycle: how to make claims and design by scenario. ACM Trans. Inf. Syst. 10, 2 (April 1992), 181–212
[11] D. Navarre, P. Palanque, R. Bastide. A Tool-Supported Design Framework for Safety Critical Interactive Systems in Interacting with computers. In: Interacting with computers, 2003, 15(3), 309-328.
[12] A. General, B. Da Silva, D. Esteves, M. Halleran, M. Liut. A comparative analysis between the mouse, trackpad and the leap motion. In: CHI Conference, 2016.
[13] K Ken Hinckley, Robert J. K. Jacob, Colin Ware, Jacob O. Wobbrock, Daniel Wigdor. Input/Output Devices and Interaction Techniques. Computing Handbook, 3rd ed. (1) 2014: 21: 1-54
[14] R. J. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. In: ACM Transactions on Information Systems (TOIS), 1991, 9(2), 152-169.
[15] Logitech MX ERGO webpage, https://www.logitech.com/en-us/shop/p/mx-ergo-wireless-trackball-mouse, last accessed February 2025.
[16] P. Majaranta, A. Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, 2014, 39-65.
[17] C. Martinie, P. Palanque, E. Bouzekri, A. Cockburn, A. Canny, E. Barboni. Analysing and demonstrating tool-supported customizable task notations. Proceedings of the ACM on Human-Computer Interaction, 3(EICS), 2019, Article 12, 26 pages. https://doi.org/10.1145/3331154
[18] V. B. Nguyen, F. Jambon, G. Calvary. Gaze-based interaction: evaluation of progressive feedback. In: Proceedings of the 26th Conference on l'Interaction Homme-Machine (IHM '14), 2014, 153–158. https://doi.org/10.1145/2670444.2670463

[19] A. M. Penkar, C. Lutteroth, G.Weber. Designing for the eye: design parameters for dwell in gaze interaction. In: Proceedings of the 24th Australian Computer-Human Interaction Conference (OzCHI '12), 2012, 479–488. https://doi.org/10.1145/2414536.2414609

[20] H. Singh, J. Singh. Object Acquisition and Selection in Human Computer Interaction Systems: A Review. In: International Journal of Intelligent Systems and Applications in Engineering, 7(1), 2019, 19–29. https://doi.org/10.18201/ijisae.2019151248.

[21] Smart Eye AI-X eye tracker webpage, https://www.smarteye.se/ai-x/, last accessed March 2025.

[22] S. Steere, E. Julien, F. Manon. The CDO: An innovative, flexible and modern Operations Control Centre for Europe's spaceport, French Guiana: Ground system architecture, resilience & operational excellence. In: Journal of Space Safety Engineering, 2024, 11, 2, 301–310. https://doi.org/https://doi.org/10.1016/j.jsse.2024.04.006.

[23] Vega-C launch vehicle webpage, https://cnes.fr/en/projects/vega-c, last accessed December 2024.

[24] A. Zaranek, B. Ramoul, H. F. Yu, Y. Yao, R. J. Teather. Performance of modern gaming input devices in first-person shooter target acquisition. In: CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14), 2014, 1495–1500. https://doi.org/10.1145/2559206.2581291.

[25] S. Zhang, Y. Tian, C. Wang, K. Wei, K. Target selection by gaze pointing and manual confirmation: performance improved by locking the gaze cursor. In: *Ergonomics*, *63*(7), 2020, 884–895. https://doi.org/10.1080/00140139.2020.1762934

[26] C. Martinie, Philippe Palanque, Eric Barboni. Principles of Task Analysis and Modeling: Understanding Activity, Modeling Tasks, and Analyzing Models. Handbook of Human Computer Interaction, Springer International Publishing, pp.1-40, 2022.

[27] C. Martinie, David Navarre, Philippe Palanque, Eric Barboni, and Sandra Steere. 2022. Engineering Operations-based Training. Proc. ACM Hum.-Comput. Interact. 6, EICS, Article 164 (June 2022), 25 pages.

[28] G. Lallai, Giovanni Loi Zedda, Célia Martinie, Philippe Palanque, Mauro Pisano, Lucio Davide Spano. Engineering Task-based Augmented Reality Guidance: Application to the Training of Aircraft Flight Procedures. Interacting with Computers. 33(1), 17-39 (2021). DOI: https://doi.org/10.1093/iwcomp/iwab007

[29] A. van Dam. 1997. Post-WIMP user interfaces. Commun. ACM 40, 2 (Feb. 1997), 63–67.

[30] J. Vermeulen, K. Luyten, E. van den Hoven, and K. Coninx. "Crossing the bridge over Norman's Gulf of Execution: revealing feedforward's true identity." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1931-1940. 2013.

[31] J. Accot, S. Chatty, and P. Palanque. "A formal description of low-level interaction and its application to multimodal interactive systems." In Design, Specification and Verification of Interactive Systems' 96: Proceedings of the Eurographics Workshop in Namur, Belgium, June 5–7, 1996, pp. 92-104. Vienna: Springer Vienna, 1996.

[32] J. Saint-Lot, J-P. Imbert, and Frédéric Dehais. 2020. Red Alert: A Cognitive Countermeasure to Mitigate Attentional Tunneling. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–6.

[33] A. Cockburn, D. Masson, C. Gutwin, P. Palanque, A. Goguey, M. Yung, C. Gris, C. Trask. "Design and evaluation of braced touch for touchscreen input stabilisation." International Journal of Human-Computer Studies 122 (2019): 21-37.

[34] P. Palanque, A. Cockburn, L. Désert-Legendre, C. Gutwin, Y. Deleris. Brace touch: a dependable, turbulence-tolerant, multi-touch interaction technique for interactive cockpits. In: Computer Safety, Reliability, and Security: 38th International Conference, SAFECOMP 2019, Turku, Finland, September 11–13, 2019, Proceedings 38 (pp. 53-68). Springer International Publishing.

[35] S. Steere, E. Barboni, C. Martinie, D. Navarre, P. Palanque, et al. Operator-centered and model-based design for critical HMIS. 3rd Ground-Based Space Facilities Symposium (GBSF 2022), Dec 2022, pp.1-9